UNIVERSITY OF CALIFORNIA
RIVERSIDE

Context-Aware Informative Sample Selection and Image Forgery Detection

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering

by

Md Jawadul Hasan Bappy

June 2018

Dissertation Committee:

Dr. Amit K. Roy-Chowdhury, Chairperson
Dr. B.S. Manjunath
Dr. Ertem Tuncel

Copyright by
Md Jawadul Hasan Bappy
2018

The Dissertation of Md Jawadul Hasan Bappy is approved:

____

____

____

Committee Chairperson

University of California, Riverside

Acknowledgments

First and foremost, I would like to thank my PhD advisor- Professor Amit K. Roy-Chowdhury, who has been always supportive of my career goal. Professor Amit has supported me not only by providing financial aid during my PhD period, but also academically and emotionally through the rough road to finish this thesis. He always guided and encouraged me to come up with new and simple solutions to complex problems. I always had the moral support and the freedom to work on interesting problems during PhD. I have learned a lot from his expertise and immense knowledge. I could not have imagined having a better advisor and mentor for my PhD study.

Besides my advisor, I would like to thank the rest of my thesis committee: Professor B.S. Manjunath, and Professor Ertem Tuncel for their continuous support developing this thesis. During my PhD, I had the opportunity to work with Professor Manjunath who is a very highly regarded researcher in image processing and computer vision. His expertise, guidance and motivation helped me develop major part of my thesis as provided in Chapter 4. Without his precious support it would not be possible to conduct this research. I am also lucky to work with Professor Tuncel. He is an excellent teacher and mentor. He was also involved in one of my research work as demonstrated in Chapter 3 where I learned a lot from his knowledge and experience in the field.

I would like to thank my fellow labmates in the Video Computing Group at UC Riverside. In particular, I would like to express my sincere gratitude to Sujoy Paul, with whom I had lot of discussions on various topics and several brainstorming spells in the lab. My special thanks to Dr. Mahmudul Hasan, Rameswar Panda, Niluthpol Chowdhury, Tahmida Binte Mahmud, Sourya Roy, and Sudipta Paul for the stimulating discussions, and for all the fun we have had in the last four years.

Last but not the least, I would like to thank my parents for their inspiration and continuous support to pursue PhD study. My gratefulness for their sacrifice can not be expressed in words. I am grateful to my sister Sharmin Badhan for her support and love. Most importantly, I wish to thank my loving and supportive wife, Farzana Rahman, who gave me perpetual inspiration. I also like to thank my mother-in law, father-in-law and sister-in-law for their guidance and motivation.

To my parents.

ABSTRACT OF THE DISSERTATION

Context-Aware Informative Sample Selection and Image Forgery Detection

by

Md Jawadul Hasan Bappy

Doctor of Philosophy, Graduate Program in Electrical Engineering
University of California, Riverside, June 2018
Dr. Amit K. Roy-Chowdhury, Chairperson

Most of the computer vision methods assume that data will be labeled and available beforehand in order to train a good recognition model. However, it becomes infeasible and unrealistic to know all the labels beforehand with the huge corpus of visual data being generated on a daily basis. In most image and video analysis tasks, selection of the most informative samples from a huge pool of training data in order to learn a good recognition model is an active research problem. Furthermore, it is also useful to reduce the annotation cost, as it is time-consuming to annotate unlabeled samples. In this thesis, we aim to design information-theoretic approaches which exploit inter-relationships between data instances in order to find informative samples in image or videos. Moreover, in recent years, the advent of high-tech journaling tools facilitates an image to be forged in a way that can easily evade state-of-the-art image tampering detection approaches. The recent success of the deep learning approaches in different recognition tasks inspires us to develop a high-confidence detection framework which can localize forged/manipulated regions in an image. Unlike semantic object segmentation where all meaningful regions (objects) are segmented, the localization of image forgery focuses only the possible tampered region which makes the problem even more challenging.

We present two distinct information-theoretic approaches for selecting samples to learn recognition models, and a deep learning based method for localizing manipulation from images. In first approach, we show how models for joint scene and object classification can be learned online. A major motivation for this approach is to exploit the hierarchical relationships between scenes and objects, represented as a graphical model, in an active learning framework. To select the samples on graph, which need to be labeled by a human, we formulate an optimization function that reduces the joint entropy of scene and object variables. The second approach we propose is motivated by the theories in data compression, which exploits the concept of typicality from the domain of information theory in order to find informative samples in videos. Typicality is a simple and powerful technique which can be applied to compress the training data to learn a good classification model. Both of the approaches lead to a significant reduction in the amount of manual labeling effort for similar or better performance when compared with a model trained with the full dataset. In the final chapter, we explore a deep learning architecture to localize manipulated regions from an image. Our proposed framework utilizes resampling features, Long-Short Term Memory (LSTM) cells, and encoder-decoder network to segment out manipulated regions from non-manipulated ones. The overall framework is capable of detecting different types of image forgeries.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In computer vision, there has been significant effort to train a good recognition model for different visual recognition tasks such as scene classification, object detection and activity recognition. Learning the classification model for these tasks requires lot of data. Most existing methods assume that data will be labeled and available beforehand in order to train the classification models. It becomes infeasible and unrealistic to know all the labels beforehand with the huge corpus of visual data being generated on a daily basis. Moreover, adaptability of the models to the incoming data is crucial too for long-term performance guarantees. Currently, the big datasets (e.g. ImageNet [39], SUN [153]) are prepared with intensive human labeling, which is difficult to scale up as more and more new images/videos are generated. So, we want to pose a question, '*Are all the samples equally important to manually label and learn a model?*'. In this thesis, we propose two different information-theoretic approaches which exploit inter-relationships between the samples in order to select informative instances to learn recognition model. Moreover, in recent years, digital image forensics has been a growing interest in diverse scientific and security/surveillance applications. With advanced image journaling tools, one can easily alter the semantic meaning of an image by exploiting certain manipulation techniques such as copy-clone, object splicing, and removal, which mislead the viewers. In contrast, the identification of these manipulations becomes a very challenging task as manipulated regions are not visually apparent. In this thesis, we also aim to develop deep learning architecture in order to recognize manipulated/forged objects from an image.

Scene classification is a challenging problem due to the severe differences in intra-class and inter-class scene categories [42]. Most of the feature-based object recognition algorithms perform poorly in the face of variability of illumination, deformation, background clutter and occlusion. As scene and objects are interrelated, the performance of both of these recognition tasks can be further improved by exploiting dependencies between scene and object deep networks. Recently, there are also a lot of interest in online adaptation of recognition models as new data becomes available. Active learning [139] has been widely used to choose a subset of most informative samples that can achieve similar or better performance than all the data being manually labeled. Most of the existing active learning approaches consider the individual samples to be independent. However, there are various tasks, such as document classification [108] and activity recognition [161], where interrelationships between samples exist. In such cases, it will be advantageous to exploit these relationships to reduce the number of samples to be manually labeled. Similar to the applications mentioned above, exploiting mutual relationships between scene and objects can yield better performance [156] than if no relationships are considered. In our first approach, we propose a novel active learning framework which exploits the mutual relationships to jointly learn scene and object classification models. Using mutual relationships between scene and objects, we can leverage upon the fact that manual labeling of one reduces the uncertainty of the other, and thus reduces labeling cost. This is achieved using an information theoretic approach that reduces the joint entropy of a graph.

Classification task such as activity recognition in videos, relies on labeled data in order to learn a recognition model. In [77], it has been shown that more labeled data do not always help a recognition model to learn better; sometimes the performance might even degrade due to noisy data points. Thus, selection of the most informative samples to train a recognition model becomes crucial. In information theory, the idea of 'typical set' is successfully applied in compression theory, which is based on the intuitive notion that not all the messages are equally important, i.e., some messages carry more information than others. By analogy, we can exploit this concept to reduce the manual labeling cost by choosing the most informative samples from a large pool of unlabeled data. Typicality allows representation of any sequence using entropy as a measure of information [34]. We present our second approach which exploits the concept of typicality from the domain of information theory in order to find informative samples for activity recognition in videos. In activity recognition, current activity is strongly correlated with previous activity sample, thus exhibits Markovian property. We assume that action samples produce a Markov chain where current sample only depends on the previous sample, and demonstrate how to utilize typicality for this scenario. Moreover, typicality based sample selection approach is computationally faster than existing graph-based approaches [9, 118, 60] that exploit the correlation between the samples. The notion of typicality can also be utilized for automatic detection of unusual or abnormal activities in videos.

Digital image forensics is an emerging important topic in diverse scientific and security/surveillance applications. With the availability of digital image editing tools, digital altering or tampering of an image has become very easy. In contrast, the identification of tampered images is a very challenging problem due to the strong resemblance of a forged image to its original one. There are certain types of forgeries such as copy-move, splicing, removal, that can easily deceive the human perceptual system. Most of the existing methods have focused on classifying whether an image is forged or not. However, there are few methods [130, 50, 17] that localize manipulated regions from an image. Some recent works address the localization problem by classifying patches as manipulated. Towards the goal of detecting and localizing manipulated image regions, we present our final framework which is based on deep learning architecture in order to localize manipulated/forged objects. The proposed network exploits resampling features, LSTM network, and encoder-decoder architectures in order to learn the pixel level localization of manipulated image regions. Given an image, we divide into several blocks/patches and then resampling features are extracted from each block. LSTM network is utilized to learn the correlation between manipulated and non-manipulated blocks at frequency domain. We utilize and modify encoder-decoder network as in [8] to capture spatial information. Each encoder generates feature maps with varying size and number. The feature map from LSTM network and the encoded feature maps from encoders are embedded before going through the decoder. We perform end-to-end training to learn the parameters of the network through back-propagation using ground-truth mask information. As deep networks are data hungry, we create lots of synthesized images to learn the base model. The proposed model shows promising results in localizing forged regions at the pixel level, which is demonstrated on different challenging datasets.

Chapter 2

Online Adaptation for Scene and Object Recognition Models

2.1 Introduction

Scene classification and object detection are two challenging problems in computer vision due to high intra-class variance, illumination changes, background clutter and occlusion. Recent efforts in computer vision consider joint scene and object classification by exploiting mutual relationships (often termed as context) between them to achieve higher accuracy. Typically, training these recognition models (e.g., scene and objects) require lot of labeled data. However, this assumption may be too strong in real-life applications. In this chapter, we aim to develop an information-theoretic framework to select the most informative samples in order to reduce the annotation cost.

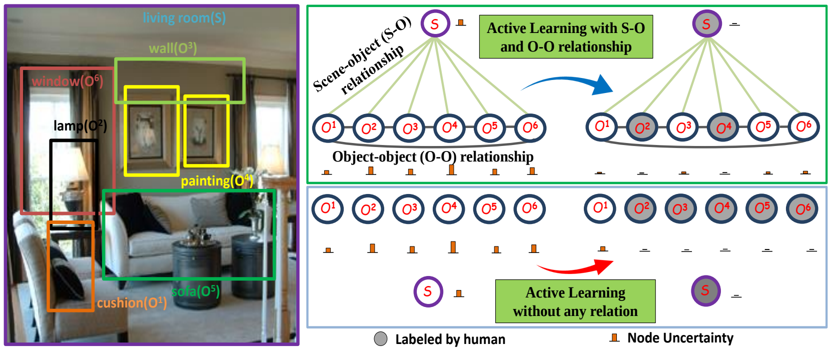

living room(S)
wall(O3)
window(O6)
lamp(O2)
painting(O4)
sofa(O5)
cushion(O1)
Scene-object (S-O) relationship
S
Active Learning with S-O and O-O relationship
O1 O2 O3 O4 O5 O6
Object-object (O-O) relationship
Active Learning without any relation
Labeled by human
Node Uncertainty

Figure 2.1: This figure presents the motivation of incorporating relationship among scene and object samples within an image. Here, scene (S) and objects ($O^1, O^2, \ldots, O^6$) are predicted by our initial classifier and detectors with some uncertainty. We formulate a graph exploiting scene-object (S-O) and object-object (O-O) relationships. As shown in the figure, even though $\{S, O^2, O^3, O^4, O^5, O^6\}$ nodes have high uncertainty, manually labeling only 3 of them is good enough to reduce the uncertainty of all the nodes if S-O and O-O relationships are considered. So, the manual labeling cost can be significantly reduced by our proposed approach.

In computer vision, researchers exploit active learning [139] to select the most informative samples to reduce manual labeling cost. In order to identify the informative samples, most active learning techniques choose the samples about which the classifier is most uncertain. Expected change in gradients [139], information gain [85], expected prediction loss [84] are some approaches used in the literature to obtain the samples for query. These approaches consider the individual samples to be independent. Recognition tasks, such as document classification [108] and activity recognition [161], share interrelationships between samples. Some active learning frameworks consider the interrelationship between samples, and exploit different contextual relations such as link information [140], social relationships [66], spatial information [81], feature similarity [100], spatio-temporal relationships [60].

We leverage upon active learning for identifying the samples to label in the problem of joint scene and object recognition. In the context of joint scene-object classification, exploiting mutual relationships between scene and objects can achieve better performance [156] than if no relationships are considered. For example, it is unlikely to find a ‘cow’ in a ‘bedroom’, but, the probability of finding ‘bed’ and ‘lamp’ in the same scene may be high. Thus gaining information about a scene can help in enhanced prediction on objects and vice versa. Previously, research in [151, 5, 148, 114] has shown how to exploit the scene-object relationships to yield better classification performance. However, these methods require data to be manually labeled and available before learning. Although there exist some works involving active learning in scene and object classification [85, 84, 83], they do not exploit the scene-object(S-O) and object-object(O-O) inter-relationships. This is critical because of the hierarchical nature of the relationships between objects and scenes. This relationship can be represented as a graphical model with the samples on the graph, which need to be labeled by a human, chosen using a suitable criterion. The labeling effort can be significantly reduced in this process - labeling a scene node in the graph can possibly resolve ambiguities for multiple object classes. This motivation is portrayed in Fig. 4.1.

Motivated by the above, we present a novel active learning framework which exploits the S-O and O-O relationships to jointly learn scene and object classification models. We exploit graphical model in order to relate scene and object samples given an image. We compute the joint entropy of the graph, which represents the total uncertainty over all the nodes. In our approach, we observe that manual labeling of one reduces the uncertainty of the other, and thus reduces labeling cost. This is achieved using an information theoretic approach that reduces the joint entropy of a graph. As presented in the figure, exploiting

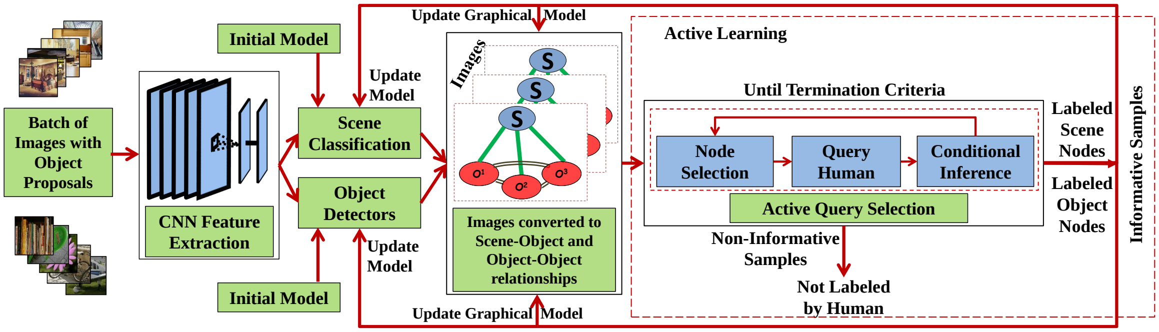


Figure 2.2: This figure presents a pictorial representation of the proposed framework. At first, initial classification models and relationship model are learned from a small set of labeled images. Thereafter, as images are available in batches, scene and object classification models provide prediction scores of scene and objects. With these scores and the relationship model, the images are represented as graphs with scene and object nodes. Then, the active learning module is invoked which efficiently chooses the most informative scene or object nodes to query the human. Finally, the labels provided by the human are used to update the classification and relationship models.

relationships between scene and objects can lead to lesser human labeling effort, compared to when relationships are not considered.

Framework Overview. The flow of the proposed algorithm is presented in Fig. 3.1. We perform two tasks simultaneously:

1. Selection of an image that contains the most informative samples (scene,objects)

2. Given an image, a sample (i.e., a node in the graph representing that image) is chosen in a way that reduces the uncertainty on other samples.

Our framework is divided into two phases. At first, we learn the initial classification models as well as the S-O and O-O relationship model with small amount of labeled data. In the second phase, with incoming unlabeled data, we first classify the unlabeled scene and object samples using the current models. Then, we represent each incoming image as a graph, where scene classification probabilities and object detection scores are utilized to represent the scene and object node potentials. S-O and O-O relations delineate the edge potentials. We compute the marginal probabilities of node variables from the inference on the graphs.

Thereafter, we formulate an information-theoretic approach for selecting the most informative samples. Joint entropy of a graph is computed from the joint distribution of scene and objects that represents the total uncertainty of an image. For a batch of data, our framework chooses the most informative samples based on some uncertainty measures (discussed in Sec. 2.3) that lead to the maximum decrease in the joint entropy of the graph after labeling. After receiving the label of a node from the human, we infer on the graph conditioned upon the known label. Due to this inference, the other unlabeled nodes gain information from the node labeled by human, which leads to a significant reduction in uncertainties of other nodes. The labels obtained in this process are used to update the scene and object classification models as well as the S-O and O-O relationships.

2.1.1 Main Contributions.

Our main contributions are as follows.

• In computer vision, most of the existing active learning methods involve learning a classification model of one type of variable, e.g., scene, objects, activity, text, etc. On the other hand, the proposed active learning framework learns scene and object classification models *simultaneously.*

• In the proposed active learning framework, both scene and object classification models take advantage of the interdependence between them in order to select the most informative samples with the least manual labeling cost. To the best of our knowledge, any previous work using *active learning to classify scene and objects together* is unknown.

• Leveraging upon the inter-relationships between scene and objects, we propose a new information-theoretic sample selection strategy along with inference on a graph based on the intuition that learning a sample reduces the uncertainties of other samples. Moreover, our framework facilitates continuous and incremental learning of the classification models as well as the S-O and O-O relationship models, thus dynamically adapting to the changes in incoming data.

2.1.2 Related Works

Scene and Object Recognition. Many of the scene classification methods use low dimensional features such as color and texture [158], GIST [87], SIFT descriptor [92] and deep feature [165]. In object detection, current state-of-the-art methods are R-CNN [53], SPP-net [63] and fast R-CNN [52]. Another promising approach in recognition tasks has been to exploit the relationships between objects in a scene using a graphical model [30], [167], [156]. A Conditional Random Field (CRF) for integrating the scene and object classification for video sequences was proposed in [151]. A model for joint segmentation, object and scene class inference was proposed in [156]. In [5], the spatial relationships between the objects within an image were exploited to compute the scene similarity score, based on which the indoor scene categories were predicted. In [148], a CRF model was constructed based on scene, object and the textual data associated with the images on the web, to label the scenes and localize objects within the image. In [160], a projection was formulated from images to a space spanned by object banks, based on which, the image was classified into different categories. In [114], a framework was developed for multiple object recognition within an image, where a conditional tree model was learned based on the co-occurrences of objects.

Active Learning. Although the above mentioned works exploit the contextual relationships, they assume that all the data are labeled and available beforehand, which is not feasible and involves huge labeling cost. Active learning has been widely used to reduce the effort of manual labeling in different computer vision tasks including scene classification [85], video segmentation [46], object detection [144], activity recognition [60], tracking [147]. A generalized active learning framework for computer vision problems such as person detection, face recognition and scene classification was proposed in [43]. They used the two concepts of uncertainty and sample diversity to choose the samples for manual labeling. Some of the common techniques to measure uncertainty for selecting the informative data points are presented in [138]. Active learning has been separately used for scene or object classification [85, 83, 144, 72], but not in their joint classification.

In [83], a framework for actively learning scene classification model was proposed, where the authors incorporated two strategies - Best vs. Second Best (BvSB) and K-centroid to select the informative subset of images. A framework based on information density measure and uncertainty measure to obtain the best subset of images for querying the human was proposed in [84]. Although their algorithm can be applied separately for both scene and object classification, they do not exploit the relationships between scene and objects. An active learning framework for object categories was proposed in [69] which considers the case where the labeler itself is uncertain about labeling an image.

In [85], the authors present an active learning framework for scene classification. In their hierarchal model, they focus on querying at the scene level, and whenever unexpected class labels are returned by the human, queries are made at the object level. Thus in their method, there exists a flow of information from the object level to the scene level. However, in our method, there is a flow of information from scene to object level and vice versa, in a collaborative manner, which paves the path for a joint scene-object classification framework.

2.2 Joint Scene and Object Model

In this section, we discuss how we represent an image in a graphical model with scene and object as hidden variables.

***A.* Scene Classification Method.** In order to represent scenes, we extract features using Convolution Neural Networks (CNN). Given an image, we get a feature vector f from the $fc7$ layer of a CNN architecture, where $f \in \Re^{4096\times 1}$. We train a linear multi-class Support Vector Machine (SVM) [24] to compute the probability of n^{th} class, $p(S = s_n|f^j)$, where f^j implies the feature vector corresponding to sample j. We denote the learned model for scene classification as $\mathcal{P}_s$. Given an image, $\varPhi_S \in \Re^N$ represents the classification score. N is the total number of scene categories considered in the experiment.

***B.* Object Detection Method.** We use R-CNN presented in [53] to detect the objects in an image. In R-CNN, we extract features from deep network for each object proposal. Then, we train a binary SVM classifier for each object category to get the probability of appearance of an object. After classifying the region we form a vector that represents the confidence scores of the binary classifiers for each category. Thus, for each p^{th} region we get $\varPhi_{O^p}$ that represents the detection score vector. Finally, we use bounding box regression method [47] for better object localization. We denote the learned model for scene classification as $\mathcal{P}_o$.

***C.* Graphical Model Representation.** In this model, two levels of nodes are used - one represents scene υ_s and other set of nodes implies detected objects υ_o. υ_o is generally represented by $\upsilon_o = \{\upsilon_{o^1}, \upsilon_{o^2}, ..\upsilon_{o^D}\}$, where D is the number of bounding boxes appearing in an image. The link between them is depicted by edges. The joint distribution of υ_s and υ_o over the CRF can be written as

$$P(\upsilon_s, \upsilon_o) = \frac{1}{Z}\ \Psi_\xi(\upsilon_s, \upsilon_o) \prod_{\substack{i,j \in D \\ i \neq j}} \Psi_\xi(\upsilon_{o^i}, \upsilon_{o^j}) \prod_{w \in \{\upsilon_s, \upsilon_o\}} \Psi_v(w) \tag{2.1}$$

where, Z is normalizing constant. $\Psi_v(.)$ and $\Psi_\xi(.)$ denote node and edge potentials.

Node Potentials. Given an image, the scene classifier $(\mathcal{P}_s)$ produces a vector that contains the probabilities of all the scene labels. From these probabilities we compute scene node potential $\Psi_v(\upsilon_s)$ as presented in Eqn. 2.2. Similarly, given an image, the object detection scores are used to model the object node potentials $\Psi_v(\upsilon_o)$ as shown in Eqn. 2.3.

$$\Psi_v(\upsilon_s) = \sum_{n \in N} \mathcal{I}(S_n)\beta_n^T\ \mathit{\Phi}_S \tag{2.2}$$

$$\Psi_v(\upsilon_o) = \sum_{p \in D}\sum_{m \in M} \mathcal{I}(O_m^p)\Omega_m^T\ \mathit{\Phi}_{O^p} \tag{2.3}$$

Here, $\mathit{\Phi}_S$ is a vector of the probability of the scene labels obtained from multi-class SVM classifier. β_n is the feature weight vector corresponding to scene label S_n and $\mathcal{I}(.)$ is the indicator function, i.e., $\mathcal{I}(S_n) = 1$ when $S = S_n$, otherwise 0. Ω_m is the weight corresponding to the detection score of the object O_m. $\mathit{\Phi}_{O^p}$ is the score vector of detecting all the objects in the p^{th} bounding box. M is the number of object Classes.

Edge Potentials. We use two type of relationships, S-O and O-O. We use co-occurrence frequencies to represent edge potential. The probability of the presence of an object in a particular scene is determined by the co-occurrence statistics. For instance, in a context of *'highway'* scene, the probability of appearance of *'car'* will be higher than *'table'* or *'chair'*. In Eqn.2.4, $\Psi_\xi(\upsilon_s, \upsilon_o)$ represents the relationship between S and O. Similarly, $\Psi_\xi(\upsilon_{o^i}, \upsilon_{o^j})$ models the O-O relations.

$$\Psi_\xi(\upsilon_s, \upsilon_o) = \sum_{p \in D} \sum_{n \in N} \sum_{m \in M} \mathcal{I}(S_n)\mathcal{I}(O_m^p)\mathit{\Phi}_\xi(S_n, O_m) \tag{2.4}$$

$$\Psi_\xi(\upsilon_{o^i}, \upsilon_{o^j}) = \sum_{m' \in M} \sum_{m \in M} \mathcal{I}(O_{m'}^i)\mathcal{I}(O_m^j)\ \mathit{\Phi}_\xi(O_{m'}, O_m) \tag{2.5}$$

$\mathit{\Phi}_\xi(S_n, O_m)$ represents the co-occurrence statistics between scene and objects. Larger value implies higher probability of co-occurrence of S_n and O_m. Here, $\mathit{\Phi}_\xi(O^i, O^j)$ is the co-occurrence [126] between the detected objects O^i and O^j. It encodes the information about how often two objects can co-occur in a scene.

Parameter Learning. The initial model parameters of the CRF model are learned from a set of annotated images, object detectors and scene classifier. Given the ground truth object bounding boxes, we use object detectors to obtain detection scores for the corresponding bounding box region. Similarly, we get the classification score from the annotated scene label. Thus, we can easily apply maximum likelihood estimation approach to learn all the parameters $\{\beta, \Omega, \mathit{\Phi}_\xi(S_n, O_m), \mathit{\Phi}_\xi(O_{m'}, O_m)\}$ in the model.

Inference of Scene and Object Labels. To compute the marginal distributions of the node and edge, we use Loopy Belief Propagation (LBP) algorithm [86], as our graph contains cycles. LBP is not guaranteed to converge to the true marginal, but has good approximation of the marginal distributions.

```
Algorithm 1: Online Learning for Scene and Object Sample Selection
    INPUTS. 1. Learned scene, object and relation models after processing
    images in Batch$_{K-1}$ : $\{\mathcal{P}_s,\mathcal{P}_o,\ \Phi_\xi(S_n,O_m)\ \&\ \Phi_\xi(O_{m'},O_m)\}$
            2. Unlabeled Batch$_K$: $\mathcal{U}$
    OUTPUTS. Learned Models after processing images in Batch$_K$: $\{\mathcal{P}_s,\mathcal{P}_o,$
    $\Phi'_\xi(S_n,O_m)\ \&\ \Phi'_\xi(O_{m'},O_m)$
    Initialize: $L_s = \{\}$ (Empty set)
    Step 1:  Compute $H(\upsilon_i)$ and $I(\upsilon_i,\upsilon_j)$ using Eqn. 2.6
    Step 2a: Compute vector $J^p = [J^p_1, J^p_2, \dots J^p_Q]$ containing the node
     uncertainties involving entropy and mutual information, for all images.
    Step 2b: Obtain vector $\hat{J}$ by concatenating the vectors $J^p$, $\forall p$, s.t.
     $H^p(V) \geq \delta$
    Step 2c: $\hat{J}_s \leftarrow sort(\hat{J})$ in descending order
    if $length(\hat{J}_s) \neq 0$ then
        Step 3a: Select nodes for manual labeling to form a set $\mathcal{M}$ using Eqn.
         2.11
        Step 3b: Query the nodes in $\mathcal{M}$ to the human
        Step 4: $L_s = L_s \cup \mathcal{M}$ (Labels provided by human)
        Step 5: Infer on the graphs conditioned on the labels provided by human
        Step 6: Update $\hat{J}_s, S$ using Steps 1 & 2a-d
    else
        Step 7: Update models $\{\mathcal{P}_s,\mathcal{P}_o,\ \Phi_\xi(S_n,O_m)\ \&\ \Phi_\xi(O_{m'},O_m)\}$ with $L_s$
```

2.3 Active Learning Framework

In the previous section, we represent an image as a graph containing υ_s and υ_o nodes. If we select a node from a graph, such that querying it will minimize the joint entropy of the graph maximally, then it means that the classifier will be able to gain maximum amount of information by labeling that node.

2.3.1 Formulation of Joint Entropy.

Consider a fully connected graph $G = (V, E)$, where V and E are the set of nodes and edges respectively. It may be noted that $V = \{S, O^1, O^2, \ldots, O^D\}$. Let $\mu_i(\upsilon_i)$ and $\mu_{ij}(\upsilon_i, \upsilon_j)$ be the marginal probabilities of the node and edge of the graph. Let υ_i and υ_j represent the random variables for nodes $i, j \in V$. In our joint scene and object classification, $i \in \{S, O^1, O^2, \ldots, O^D\}$ as discussed in Sec. 2.2. The node entropy $H(\upsilon_i)$ and mutual information $I(\upsilon_i, \upsilon_j)$ between a pair of nodes are defined as,

$$H(\upsilon_i) = \mathbb{E}[-\log_2 \mu_i(\upsilon_i)] \qquad I(\upsilon_i, \upsilon_j) = \mathbb{E}[\log_2 \frac{\mu_{ij}(\upsilon_i, \upsilon_j)}{\mu_i(\upsilon_i)\mu_t(\upsilon_j)}] \tag{2.6}$$

Considering Q nodes in the graph, its joint entropy can be expressed as,

$$\begin{aligned} H(V) &= H(\upsilon_1) + \sum_{i=2}^{Q} H(\upsilon_i | \upsilon_1, \ldots, \upsilon_{i-1}) \\ &= H(\upsilon_1) + \sum_{i=2}^{Q} \Big[H(\upsilon_i) - I(\upsilon_1, \ldots, \upsilon_{i-1}; \upsilon_i) \Big] \end{aligned} \tag{2.7}$$

using $I(\upsilon_1, \ldots, \upsilon_{i-1}; \upsilon_i) = H(\upsilon_i) - H(\upsilon_i | \upsilon_1, \ldots, \upsilon_{i-1})$. Again, using the chain rule, $I(\upsilon_1, \ldots, \upsilon_{i-1}; \upsilon_i) = \sum_{j=1}^{i-1} I(\upsilon_j; \upsilon_i | \upsilon_1, \ldots, \upsilon_{j-1})$, Eqn. 2.7 becomes

$$H(V) = \sum_{i=1}^{Q} H(\upsilon_i) - \sum_{i=2}^{Q} \sum_{j=1}^{i-1} I(\upsilon_j; \upsilon_i | \upsilon_1, \ldots, \upsilon_{j-1}) \tag{2.8}$$

It becomes computationally expensive to compute the conditional mutual information, as the number of node increases [157]. As we consider only pair-wise interactions between S-O and O-O, we approximate the conditional mutual information $I(\upsilon_j; \upsilon_i | \upsilon_1, \ldots, \upsilon_{j-1}) \approx I(\upsilon_j; \upsilon_i)$.

Thus, the joint entropy of the graph can be approximated as,

$$H(V) \approx \sum_{i=1}^{Q} H(\upsilon_i) - \sum_{i=2}^{Q} \sum_{j=1}^{i-1} I(\upsilon_j; \upsilon_i) = \sum_{i \in V} H(\upsilon_i) - \sum_{(i,j) \in E} I(\upsilon_i; \upsilon_j) \tag{2.9}$$

This expression is actually exact for a tree, but approximate for a graph having cycles. The approximation leads to the expression of joint entropy in Eqn. 2.9, which is similar to the joint entropy expression in Bethe method [157].

2.3.2 Informative node selection.

In our problem, an image is represented by a graph having several nodes with two types of hidden variables υ_s and υ_o. So, we require not only to find the most informative image but also need to choose the node to be manually labeled. If we manually label a node, then we assume that there is no uncertainty involved in that node. Thus, after labeling a node υ_i with the label l, the node entropy becomes zero, i,e. $H(\upsilon_i = l) = 0$.

Let $H^p(V)$ be the the joint entropy of image p which can be computed using Eqn. 2.9. We query the node such that $H^p(V)$ is maximally reduced after labeling the node and inferring the graph conditioned on the new label. Then, after labeling υ_i, we find the optimal node q of image p to be queried as,

$$q^* = \arg\max_q \left[H^p(\upsilon_q) - \frac{1}{2} \sum_{j \in \mathcal{N}(q)} I^p(\upsilon_q, \upsilon_j) \right] \tag{2.10}$$

where $\mathcal{N}(q)$ represents the neighbor nodes of q. For simplicity, let us define the uncertainty associated with node q of image p as $J_q^p = H^p(\upsilon_q) - \frac{1}{2} \sum_{j \in \mathcal{N}(q)} I^p(\upsilon_q, \upsilon_j)$ where the joint entropy for an image p is $H^p(V) = \sum_{q=1}^{n} J_q^p$ from Eqns. 2.9 and 2.10. From Eqn. 2.10, we choose the node to query, which has the maximum uncertainty considering not only the node entropy but also the mutual information between the nodes. Next, we explain how to choose a set of nodes from a batch of images.

2.3.3 Simultaneous Image and Node Selection.

We query the nodes of image p only if its joint entropy $H^p(V) \geq \delta$, where δ is a threshold. Since we have the information about all the node uncertainties of all images, we can perform multiple queries across multiple different images such that the learner can learn faster and more efficiently. We consider that there is no relation between the images, thus the conditional inference on one image is independent of the other images. Thus, graphs of different images can be conditionally inferred in a parallel manner.

Let, a vector, $J^p = [J_1^p, J_2^p, \ldots, J_Q^p]^T$ contain the uncertainty associated with Q (dependent on the image) nodes for an image p. Consider another vector, $\hat{J} = [J^1 \; J^2 \ldots J^P]^T$ which is obtained after concatenating all the vectors J^p for P images, whose joint entropy is higher than threshold δ. We sort the vector $\hat{J}$ in descending order to obtain a new vector $\hat{J}_s$. Then, we perform multiple queries based on $\hat{J}_s$, which contain uncertainty of nodes from multiple images of a batch. For each image, we choose the node appearing first in $\hat{J}_s$ for labeling. We perform conditional inference with the new labels in a parallel manner over all the images. The $\hat{J}_s$ vector is again obtained using the updated uncertainties of the nodes and the process is repeated until $H^p(V) \leq \delta, \forall p$. It may be noted that P decreases or at least remains same in succeeding iterations, because nodes belonging to images attaining joint entropy less than δ are not queried and thus not included in $\hat{J}_s$. Inference reduces the uncertainty on other nodes of the same image.

As uncertainty of nodes decreases, joint entropy is also reduced. Consider a matrix S having dimension $N_n \times 2$, where N_n is the total number of nodes of all images in the batch. The first and second columns of S contain the node index of a graph (image) and the image index respectively. The order in which the elements of S are populated is the same as that of $\hat{J}_s$. We refrain from choosing more than one node per image in each iteration because labeling one node can help the other nodes attain a better decision after inference. The set of nodes $\mathcal{M}$, chosen for labeling in each iteration can be expressed as,

$$\mathcal{M}^* = \underset{\substack{\mathcal{M} \\ s.t. |\mathcal{M}|=P \\ S^{i,2} \neq S^{j,2}, i,j \in \mathcal{M}}}{\arg\max} \sum_{k \in \mathcal{M}} \left[\hat{J}_s \right]_k \tag{2.11}$$

where $\left[\hat{J}_s\right]_k$ denote the k^{th} element of $\hat{J}_s$ and $S^{i,m}$ denote the i^{th} row and m^{th} column of S, where $m \in \{1, 2\}$. All the steps of active learning are shown in Algorithm 1. The first column of S is used to identify which node of an image should be labeled. To summarize Eqn. 2.11, the optimal set $\mathcal{M}$ can be obtained by choosing one node which has the highest entropy from each image.

Classifier Update. To classify scene and objects, we use a linear support vector machine (SVM) classifier. The probability of predicted label can be defined as $\hat{y} = w^T f(x) + b$, where $f(x)$ is the feature of scene or objects and w, b are parameters that determine the hyperplane between two classes. We use soft margins formulation presented in [24] to find the solution of w, b. The solution can be found by optimizing, $\frac{1}{2}w^2 + C\sum_1^n \epsilon_i$ subject to $y_i(w^T f(x_i) + b) \geq (1 - \epsilon_i)$ and $\epsilon_i \geq 0$ for all i samples, where ϵ_i is the slack variable.

Edge Weight Update. We update the co-occurrence statistics with new manually labeled data as presented in Eqns. 2.4 and 2.5. lets denote them by $\Phi'_\xi(S_n, O_m)$ and $\Phi'_\xi(O_{m'}, O_m)$. The updated co-occurrence matrix will be $[\Phi_\xi(S_n, O_m)]_{t+1} \leftarrow [\Phi_\xi(S_n, O_m)]_t + \Phi'_\xi(S_n, O_m)$ and $[\Phi_\xi(O_{m'}, O_m)]_{t+1} \leftarrow [\Phi_\xi(O_{m'}, O_m)]_t + \Phi'_\xi(O_{m'}, O_m)$, where the subscript $t+1$ indicates the edge potentials after t updates.

2.4 Experiments

In this section, we provide experimental analysis of our active learning framework for joint scene and object recognition models on three challenging datasets. For convenience, we will use terms 'inter-relationship' and 'contextual relationship' to denote scene-object and object-object relationship.

Datasets. In our experiments, we use SUN [29], MIT-67 Indoor [125] and MSRC [104] datasets in order to analyze scene classification and object recognition performance and compare our results. These datasets are appropriate as they provide rich source of contextual information between scene and objects. In SUN dataset, we choose 125 scene classes and 80 object categories to evaluate scene classification and object detection performance, as those contain annotation for both scene and objects. MIT-67 indoor [125] dataset consists of 67 indoor scene categories with large varieties of object categories. For MSRC [104] dataset, we evaluate our results comparing with the ground truth which is available in [156].

Experimental Setup. We use a publicly available software- '*UGM Toolbox*' [136] to infer the node and edge belief in image graphs. We use pre-trained model *'VGG net'* [165] which is trained on 'places-205' dataset to extract the scene features from CNN. For object recognition, we use the model as presented in [52].

In our online learning process, we perform 5 fold-cross validation, where one fold is used as testing set and the rest are used as training set. We divide the training set into 6 batches. We assume that human-labeled samples are available in the first batch and we use it to obtain the initial S and O classification models and the S-O and O-O relations. It might be possible that we do not have all the classes for scene and objects in the first batch. So, new classes are learned incrementally as batches of data come in. Now, with current batch of data we apply our framework to choose the most informative samples to label and then, update the classification and relationship models with newly labeled data. Finally, we compute our recognition results on the test set with each updated models.

Evaluation Criterion. In order to train the object detectors, we first choose positive and negative examples. We apply standard hard negative mining [47] method to train the binary SVM. We calculate the average precision (AP) of each category by comparing with the ground truth. Precision depends on both correct labeling and localization (overlap between object detection box and ground truth box). Let the computed bounding box of an object be O_b and the ground truth box be G_b, then the overlap ratio, $OR = \frac{O_b \cap G_b}{O_b \cup G_b}$. $OR \geq 0.5$ is considered as correct localization of an object. Before presenting our results, we define all the abbreviations that will be used hereafter

◇ **SOAL:** proposed scene-object active learning (SOAL) as discussed in Sec. 2.3.

◇ **Bv2B:** Best vs Second Best active learning strategy proposed in [83].

◇ **IL-SO:** Incremental learning (IL) approach presented in [59] is implemented for scene and object (SO) classification.

◇ **No Rel:** No relation is considered between scene and objects.

◇ **S-O Rel:** Only S-O relations are considered but not O-O relations.

◇ **S-O-O Rel:** Both S-O and O-O relationships are considered.

◇ **All+S-O:** All samples with S-O relations are considered.

◇ **All+S-O-O:** All samples with both S-O and O-O relations are considered.

◇ **All+No Rel:** All samples without any relation are considered.

◇ **SO+All:** All samples in batch are considered for scene and object classification with S-O-O relationship.

◇ **NL, AL:** NL implies no human in the loop, i.e., we do not invoke any human to learn labels. AL denotes active learning. For example, S-NL+O-AL means scene nodes are not queried but object nodes are queried..

Experimental Analysis. We perform the following set of experiments - 1. Comparison with other active learning methods, 2. Comparison of the baselines with different S-O and O-O relations, 3. Comparison against other scene and object recognition methods, and 4. Recognition performance of scene and object models while labeling either scene or object.

2.4.1 Comparison with Other Active Learning Methods.

In Figs. 3.5(a,b,c) and 2.4(a,b,c), we compare our active learning framework with some existing active learning approaches- Bv2B [83], Random Selection, Entropy [41] and IL-SO [59]. In the case of random selection, we pick the samples with uniform distribution. For Bv2B,Entropy and IL-SO, we implement the methods to select the informative samples for scene and objects. The feature extraction stages are the same as ours. We observe that our approach outperforms other methods by a large margin in selecting the most informative samples in both scene and object recognition.

2.4.2 Is Contextual Information Useful in Selecting the Most Informative Samples?

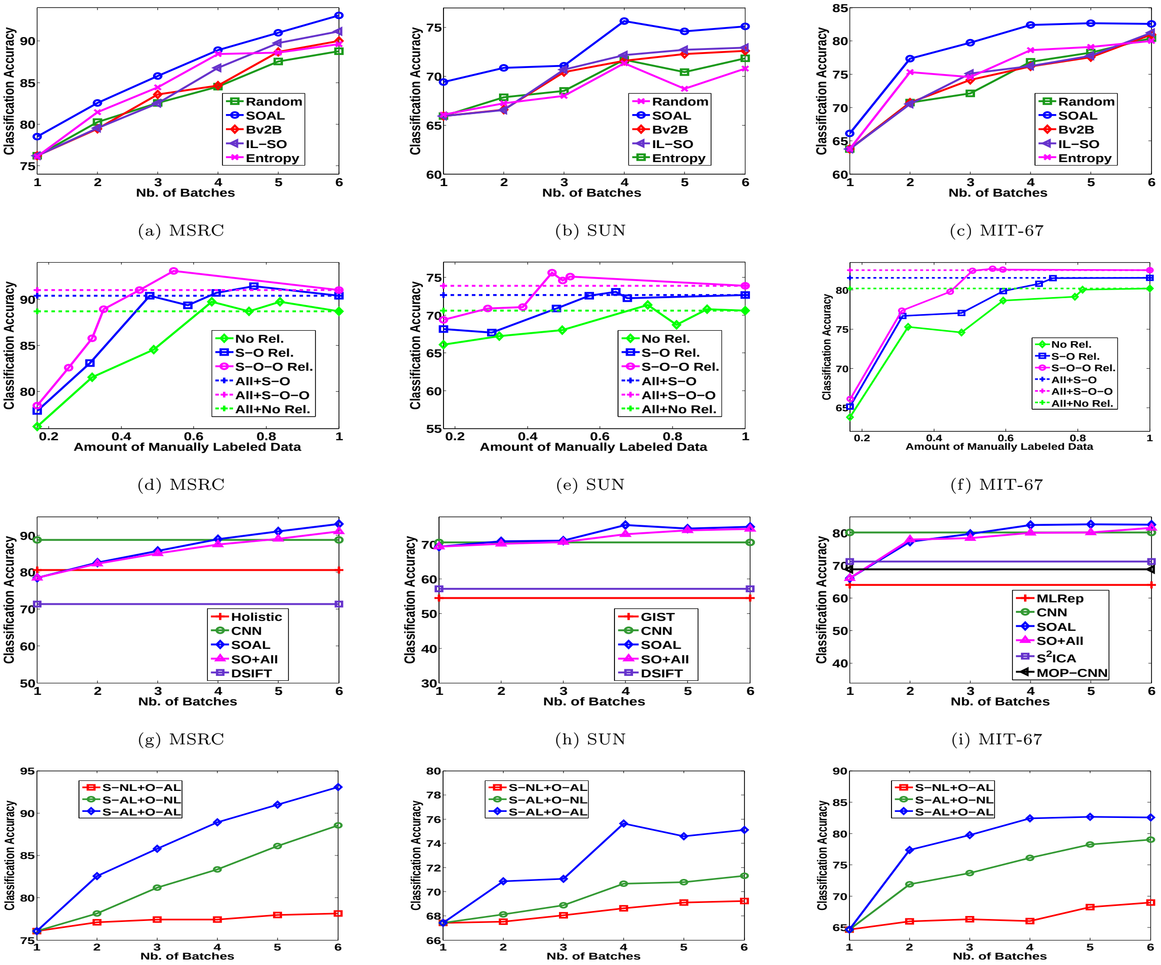
We conduct an experiment that implements our proposed active learning strategy by exploiting different set of relations of scene (S) and objects (O). Figs.3.5(d,e,f) and 2.4(d,e,f) show the plots for S and O respectively on three datasets. It is noticed that the highest accuracy is yielded by S-O-O Rel (proposed), followed by S-O Rel and No-Rel in scene classification as well as in object recognition. This brings out the advantage of exploiting both S-O and O-O relations in actively choosing the samples for manual labeling. Moreover, the manual labeling cost is significantly reduced when we consider more relations. It may also be noted that our proposed framework achieves similar or even better performance by only choosing a smaller subset of training data than building a model with full training set for both scene and objects. For scenes, this subset is **35**%, **30**% and **42**% of whole training set on MSRC, SUN and MIT datasets respectively. Similarly, for objects, we require only **39**%, **61**%, **60**% of whole training set to be manually labeled on these three datasets.

2.4.3 Comparison against other Scene and Object Classification Methods.

We also compare our S and O classification performance with other state-of-the-art S and O recognition methods. For scene, we choose Holistic [156], CNN [165], DSIFT [92], MLRep[40], S^2ICA [62] and MOP-CNN [56]. Similarly, we compare against Holistic [156], R-CNN [53], DPM [47] for object detection performance. Holistic approach exploits interrelationship among S and O using graphical model. We also compare with SO-All. From Figs. 3.5(g,h,i) and 2.4(g,h,i), we can see that our proposed framework outperforms the other state-of-the-art methods.

2.4.4 How does scene and object sample selection affect classification score of each other?

We perform an experiment to observe how S and O recognition performs, when we implement active sample selection of either scene or object nodes and exploit S-O and O-O relationships to improve the decisions of the other type of nodes. The results are shown in Figs. 3.5(j,k,l) and 2.4(j,k,l). Let us consider the first scenario (S-NL+O-AL) where we perform AL on the O nodes but use relationships to update the classification probabilities of the S node. We use the first batch to learn the S and O models, but thereafter query to label only object nodes and not scene nodes. The relationship models are updated based on



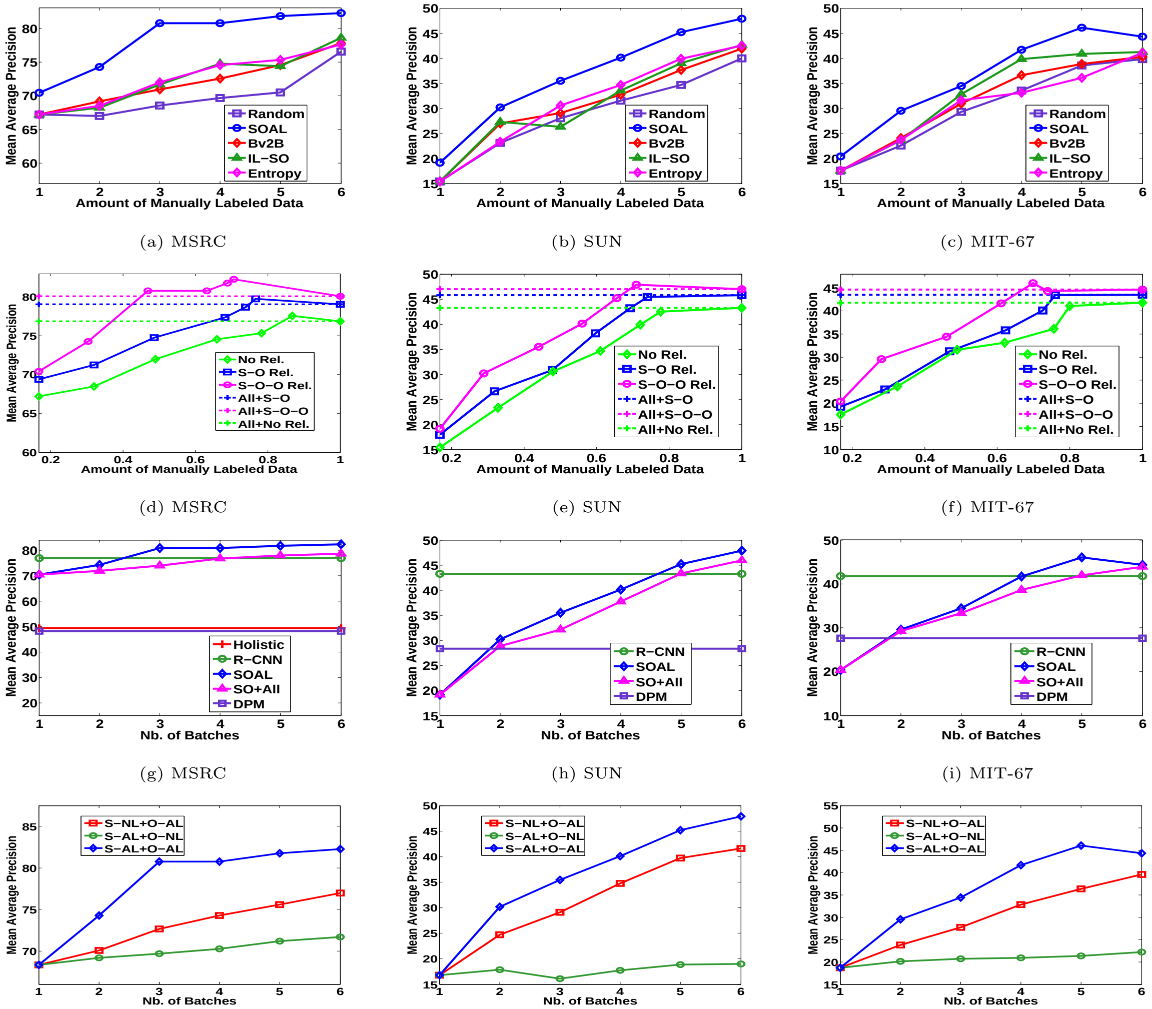
(a) MSRC (b) SUN (c) MIT-67

(d) MSRC (e) SUN (f) MIT-67

(g) MSRC (h) SUN (i) MIT-67

(j) MSRC (k) SUN (l) MIT-67

Figure 2.3: In this figure, we present the scene classification performance for three datasets- MSRC [104], SUN [153] and MIT-67 Indoor [125] (left to right). Plots (a,b,c) present the comparison of SOAL (proposed) against other state-of-the-art active learning methods. Plots (d, e, f) demonstrate comparison with different contextual relations. Plots (g,h,i) demonstrate the comparison of other scene classification methods. Plots (j,k,l) show the classification performance by utilizing our active learning framework either on scene or objects and both. Please see the experimental section for details. Best viewable in color.



(a) MSRC

(b) SUN

(c) MIT-67

(d) MSRC

(e) SUN

(f) MIT-67

(g) MSRC

(h) SUN

(i) MIT-67

(j) MSRC

(k) SUN

(l) MIT-67

Figure 2.4: In this figure, we show the object detection performances on MSRC [104], SUN [153] and MIT-67 Indoor [125] (left to right). Plots (a, b, c) present the comparison of SOAL with other state-of-the-art active learning methods. Plots (d, e, f) demonstrate comparison with different graphical relations. Plots (g, h, i) present the comparison of other object detection methods. Plots (j,k,l) show the detection performance by implementing our active learning framework either on scene or objects and both. Please see the experimental section for details. Best viewable in color.

the confidence of scene classifier and manual labeling of the objects obtained from a human annotator. With each update on context model, scene classification accuracy goes up even though the scene classification model is not updated. Similarly, the second scenario involves manual labeling of only S nodes but not O nodes. In this scenario, we do not consider O-O relationships. We can not rely on confidence of object classifiers to model O-O relations as it might provide wrong prediction of object labels. However, involvement of human in both scene and objects makes the sample selection even more efficient and outperforms all the scenarios mentioned above. As shown in Figs. 3.5(j,k,l) and 2.4(j,k,l), S-AL+O-AL achieves better performance than S-AL+O-NL by approximately **4-5**% and **4.5-5.5**% in both scene and objects on three datasets.

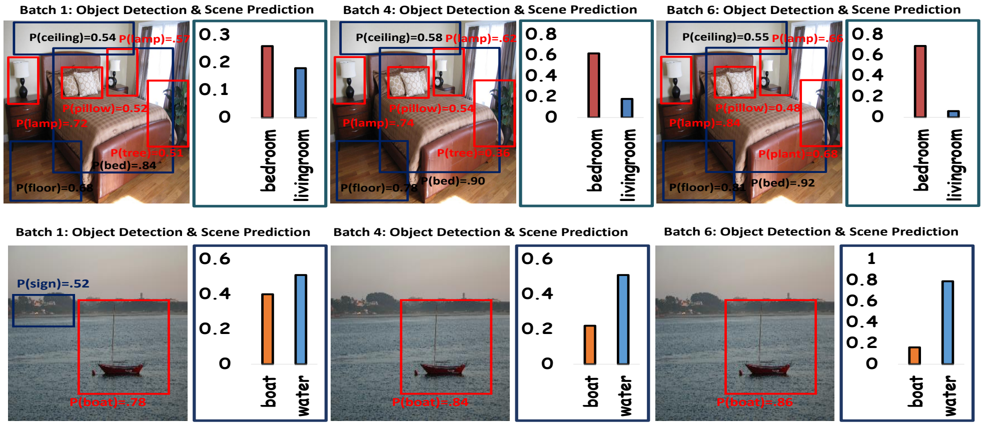


Figure 2.5: Scene prediction and object detection performance on test image with updated model learned from the data of 1^{st}, 4^{th} and 6^{th} batch.

2.4.5 Some Examples of Active Learning (AL) Performance.

We provide some examples of scene prediction and object detections as shown in Fig. 2.5. Here, scene prediction and detections are changing as models are learned over samples from each batch. Scene and object models are updated continuously with upcoming batch of data using our AL approach. With each improved model from the batch of data, classifiers become more confident in predicting scene and object labels on test image. More such examples are provided in the supplementary material.

2.5 Conclusions

In this chapter, we proposed a novel active learning framework for joint scene and object classification exploiting the interrelationship between them. We exploit the scene-object and object-object interdependencies in order to select the most informative samples to develop better classification models for scenes and objects. Our approach significantly reduces the human effort in labeling samples. We show in the experimental section that with only a small subset of the full training set we achieve better or similar performance compared with using full training set.

Chapter 3

Exploiting Typicality for Selecting Informative Samples in Videos

3.1 Introduction

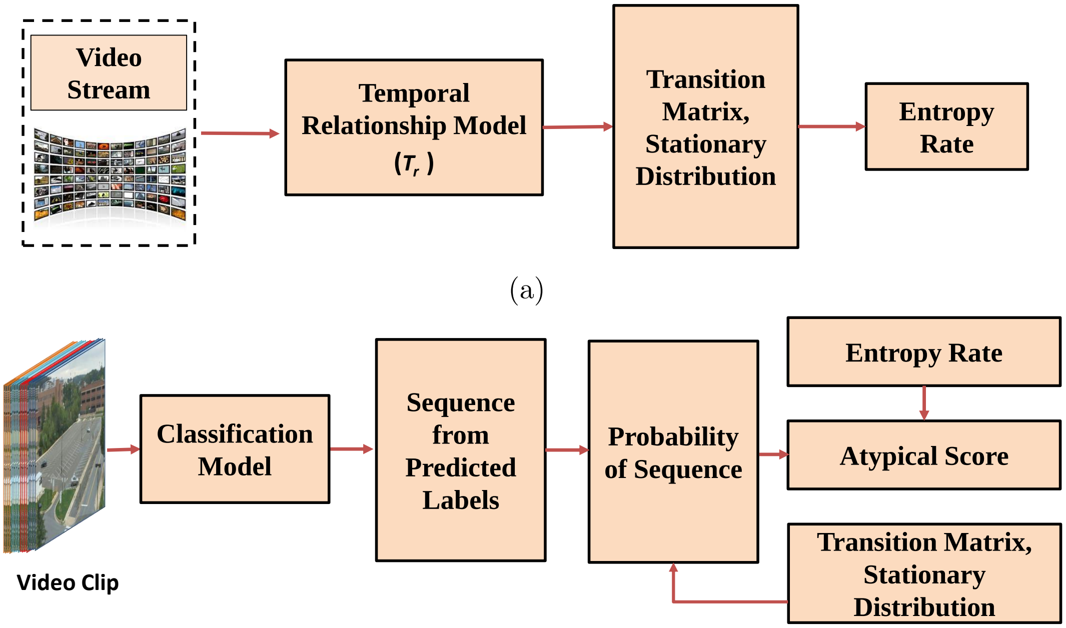
In most video analysis task, one of the challenges is to learn a good classification model from a set of labeled examples. Today we live in a time where we have instant access to huge amount of visual data from online sources such as Google, Yahoo, Bing and Youtube. It becomes infeasible to label all the unlabeled samples as it is very costly and time consuming. Moreover, it is not always true that more labeled data can help a classifier to learn better; in fact, it may as well confuse the classifier [77]. Also, the adaptability of recognition models is unavoidable in order to achieve good classification performance that is robust to concept drift. As a result, selection of the most informative samples [139] becomes critical and has drawn significant recent attention from the vision community in order to train recognition models [138, 84]. Furthermore, automatic detection of unusual or abnormal activities is an area of significant interest in diverse video analysis applications. We address both these problems in this chapter. We present *an information-theoretic approach for obtaining a subset of informative samples to learn a good classification model for activity recognition, and for identifying anomalous/irregular activities in videos.*

In computer vision, the selection of informative samples [139] has been widely used to reduce the manual labeling effort for annotation task and to train a good recognition model. Most of the sample selection methods devise a sample-wise informativeness utility score based on which the samples are selected for manual labeling [139, 85, 84]. However, they are highly dependent on classifier uncertainty or diversity in the feature space. Furthermore, the aforementioned approaches consider the individual samples to be independent. Recent works [60, 9, 10, 118] exploit the inter-relationships (or contextual information) between samples in order to reduce the number of labeled samples to train the recognition models with applications including activity recognition, scene and object classification, document classification, etc. Most of these approaches involve graphical models to exploit the interrelationships between the samples, where inference and joint entropy computation becomes intractable in the case of acyclic graphs and requires simplifying assumptions. Moreover, these methods introduce high computational complexity at the inference step as the number of nodes increases.

The analysis of abnormal activities in videos has been of growing interest in security and surveillance applications. Most of the anomaly detection methods [121, 28, 166] train a model to learn the patterns of normal activities and consider an activity as abnormal whose pattern is deviated from the normal activities. Some methods [134, 97, 32] exploit local statistics of low-level features, local spatio-temporal descriptors, and bag-of-word approach to detect anomalies in videos. Recent efforts [166, 58] in anomaly detection consider interrelationship between the activities in identifying abnormal activities. In [58], temporal regularity patterns are learned from the normal activities in order to detect unusual activity. In this work, we introduce a new way of measuring the irregularity by utilizing temporal relationship between activities to detect anomalies in video.

In this chapter, we explore whether information theoretic ideas that have been very successfully applied in data compression can be used to identify the most informative samples to build a recognition model. We leverage upon the concept of typicality for this purpose. According to the theory, there is a set of messages for which the total probability of any of its members occurring is close to one, which is referred as typical set of messages. Now, we ask how can we exploit this approach to select the most informative samples, which will be manually labeled, and classifiers designed on this subset can then be applied to the entire dataset. The concept of typical set is developed on the basis of asymptotic equipartition property of sample realization of a random variable, such that a sequence of its realization is highly likely to belong to the typical set. This concept can be utilized for informative subset selection, with the labels or a group of labels of samples being a random variable. A sequence not belonging to the typical set may be termed as informative as it does not follow the distribution of the random variable learned from the previously labeled instances. For example, in activity recognition, different activities may be temporally connected, e.g., a person opening a car trunk followed by the person carrying an object. If a different set of semantic entities appear in a particular scene, then the atypical score, computed based on the deviations in typicality, would be high and will be identified as informative. Thus, the natural interactions between semantic entities can prove to be a rich source of information in order to identify informative samples for applications like active learning, and anomaly detection.

Our previous work [10] showed preliminary results employing the concept of typicality on joint classification tasks, e.g, scene-object for images. In this work, we extend this idea more thoroughly for a range of computer vision problems in videos, such as selection of informative samples for training a model, and anomaly detection in videos. Moreover, [10] employs typicality by utilizing information flow from scene or activity to objects in a joint classification scenario, by conditioning on the former. However, it does not deal with the scenario where the scene or activity information is not known precisely. In activity recognition, the current activity may be strongly correlated with the previous activity sample, and can be represented as Markovian. In this work, we assume that action samples produce a Markov chain where the current sample only depends on the previous sample, and demonstrate how to utilize typicality for this scenario. We design an utility function which depends on the length of a sequence (please see Sec. 3.3 for more details). Moreover, we show that typicality based sample selection approach is computationally faster than existing graph-based approaches [9, 118, 60] that exploit the correlation between the samples. From the experimental results, we observe that proposed approach outperforms other state-of-the-art methods by large margin to reduce the manual labeling cost. The atypical score can also be applied to detect abnormal activities in videos (Sec. 3.3.4).


Video Stream
Temporal Relationship Model (τ_r)
Transition Matrix, Stationary Distribution
Entropy Rate
Video Clip
Classification Model
Sequence from Predicted Labels
Probability of Sequence
Entropy Rate
Atypical Score
Transition Matrix, Stationary Distribution

(a)

(b)

Figure 3.1: The figures present how typical set can be applied in vision problems to compute an atypical score. (a) represents training phase where we learn entropy rate by computing transition matrix and stationary distribution (please see Sec. 3.3.2). In (b), a sequence is generated from the prediction of activity labels given a test video. Then, the probability of the sequence is calculated from the transition matrix and stationary distribution which are learned during the training phase. Finally, we compute an atypical score for each sample in a video.

3.1.1 Framework Overview

Fig. 3.1 presents the overview of our proposed method. We can divide the overall process into two phases: (a) training phase, and (b) testing phase.

Activities in a video are represented as a Markov chain where the current activity depends on previous activity only. During the training phase, we learn the recognition model ($\mathcal{M}$) and the temporal relationship model (T_r). $\mathcal{T}_r$ could be a simple co-occurrence statistic that captures the correlation between two consecutive activities. We learn transition matrix and stationary probability using this temporal co-occurrence. We compute entropy rate required to define a typical set, details of which are provided in Sec. 3.3.

At test phase, a video clip is fed into the classifier $\mathcal{M}$. $\mathcal{M}$ provides predicted labels with a confidence score. We form a sequence from the predicted labels obtained from $\mathcal{M}$ and compute uncertainty (please see details in Sec. 3.3) of the sequence. We compare this uncertainty with the entropy of source distribution obtained from T_r in order to compute the atypical score. We can also calculate entropy from the distribution of predicted scores for each sample using $\mathcal{M}$. With this uncertainty score and atypical score, we formulate an optimization function to choose the most informative set of samples to be labeled manually by a human annotator. We also used the atypical score to detect anomalies in videos.

We applied the proposed approach to two applications- (a) informative sample selection, and (b) anomaly detection. For the first scenario, we present our approach from the perspective of batch mode active learning, where the goal is to select the most informative samples to update the recognition model in an online setting where unlabeled data are coming continuously in batches. By solving the optimization function mentioned above, we can find informative samples which will be considered for manual labeling. With these newly labeled samples, $\mathcal{M}$ and T_r are updated. In this process, we intend to achieve similar performance with the model which is trained on all the samples (100% manual labeling). In anomaly detection, we consider whole training set to understand the nature of normal activities. We learn the typical model and recognition model. Given a test sample, we set a threshold on atypical score to determine whether an activity samples is abnormal or not.

3.1.2 Contributions

Our *major contributions* are as follows.

• In this work, we present a new approach to compute an atypical score by exploiting the concept of 'typical set' from information theory. We employ our strategy on a wide range of computer vision applications such as activity recognition and anomaly detection in videos.

• Unlike [10], where the variables in a sequence are independent, we show how the concept of 'typical set' can be applied to temporally dependent variables in computer vision problems. We demonstrate our strategy on videos instead of images as presented in [10].

• We perform rigorous experimentation on two scenarios- (1) sample selection for activity classification, (2) detection of abnormal activities. Our framework on sample selection outperforms state-of-the-art methods significantly in reducing the manual labeling cost while achieving same recognition performance compared with a model trained on all the samples. We also demonstrate the usefulness of the method in finding anomalies in videos.

3.2 Related Work

In this section, we will briefly discuss the related work on visual recognition task, sample selection, anomaly detection, and typicality.

Visual Recognition Task. The proposed framework applies to work in activity classification. In [122], the paper surveys state-of-the-art feature based activity recognition. Some promising approaches in computer vision use context model [30, 60] on top of recognition model in order to achieve higher accuracy. In [60], spatio-temporal relationship and co-occurrence statistics have been utilized in order to recognize activities in video. Most of the context based approaches exploit conditional random field (CRF) to interrelate the samples, which become computationally expensive as nodes in the graph increases. Recently, various deep learning based models have been presented in [117, 159] for activity classification. These frameworks show promising performance in recognizing activities.

Sample Selection Methods. Some of the state-of-the-art sample selection approaches are expected change in gradients [139], information gain [85], expected prediction loss [84], and expected model change [70] to obtain the samples for querying. Some of the common techniques to measure uncertainty for selecting the informative samples are presented in [138, 84]. Along with classifier uncertainty, diversification in the chosen samples is introduced by using k-means [83] or sparse representative subset selection [43]. In [83], the authors incorporated two strategies - best vs. second best and K-centroid to select the informative subset. The afore-mentioned approaches consider the individual samples to be independent. Recent advances [9, 118, 60] in active learning incorporate contextual relationships to reduce manual labeling cost without compromising recognition performance.

Most of these approaches involve graph-based models where the belief is propagated through nodes using inference algorithm. These approaches might be computationally expensive as the number of nodes increases.

Anomaly Detection. Several works [166, 155] have exploited semantically meaningful activities in order to detect anomalies. A comprehensive review of anomaly detection is provided in [121]. [28] presents a hierarchical framework for identifying local and global anomalies utilizing hierarchical feature representation and Gaussian process. In [162], the authors present a method that exploits Locality Sensitive Hashing Filters (LSHF), which hashes normal activities into multiple feature buckets. [74] proposes a space-time Markov Random Field (MRF) model to identify abnormal activities in videos. Some works [154, 166] exploit spatio-temporal context in order to detect anomalous activities. In [129], the authors present an approach that learns both dominant and anomalous behaviors in videos of different spatio-temporal complexity. In anomaly detection, deep learning based approaches such as sparse auto-encoder [132] and fully convolutional feed-forward network [58] are also utilized.

Typicality. The concept of 'typical set' [101] has widely been used in compression theory as it demonstrates a theoretical justification for compressing data. Recent works [89, 115] exploit typical set in applications like multi-terminal source coding, and multiple access channel. In [65], the authors define atypicality as the deviation of the information from average. Then, it is applied in universal source coding and a number of real world datasets. In computer vision, the term 'typicality' is mentioned in some research papers for several tasks such as category search [106], object recognition [133], and scene classification [146]. However, they do not exploit the notion of information-theoretic *typical set.* In [10],

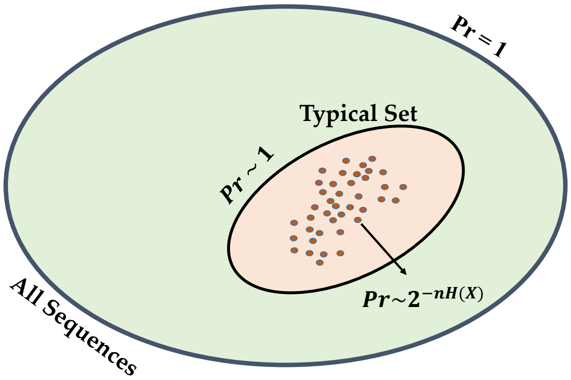


Figure 3.2: This figure presents the idea of typical set of sequences used in information theory.

a novel active learning method was proposed exploiting the theory of *typical set*. In this chapter, we extend the work presented in [10] by demonstrating its generalizability across a variety of computer vision problems.

3.3 Typicality and Its Application in Videos

In information theory, a typical set represents a set of sequences drawn from an i.i.d distribution, whose total probability of occurrence is close to one as shown in Fig. 3.2. A sequence can be categorized into either typical or atypical, depending on whether it belongs to the typical set or not. There are two kinds of typicality, namely, weak and strong. In this problem, we focus on weak typicality to develop our sample selection framework. Next, we will briefly show the concept of weak typicality and then, demonstrate how typicality can be used in different computer vision tasks.

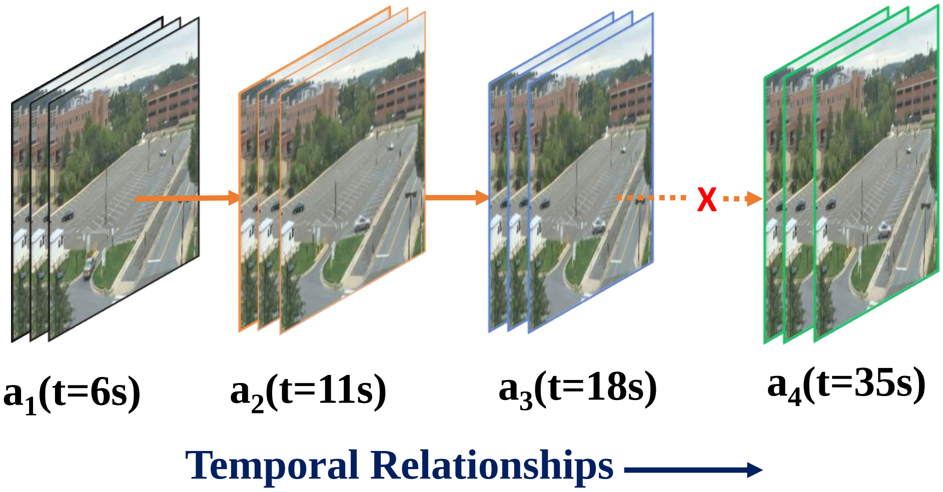


Figure 3.3: The figure presents how different activities in a video share temporal relations. Here, the temporal link between a_3 and a_4 is discarded due to long time interval.

3.3.1 Typicality in Information Theory

Let us consider $\boldsymbol{x}^n$ to denote a sequence $x_1, \ldots, x_n$ drawn from an i.i.d distribution $P_{X^n}(.)$, whose empirical entropy can be expressed as,

$$\begin{aligned} -\frac{1}{n}\log_2 P_{X^n}(\boldsymbol{x}^n) &= -\frac{1}{n}\log_2 \prod_{i=1}^{n} P_{X_i}(x_i) \\ &= -\frac{1}{n}\sum_{i=1}^{n}\log_2 P_{X_i}(x_i) \end{aligned} \quad (3.1)$$

By the weak law of large numbers Eqn. 3.1 can be written as

$$-\frac{1}{n}\sum_{i=1}^{n}\log_2 P_{X_i}(x_i) \rightarrow E[-\log_2 P_{X^n}(\boldsymbol{x}^n)] = H(X) \quad (3.2)$$

Definition. A set of sequences with probability distribution $P_{X^n}(.)$ can be considered as weakly typical set if it satisfies the following criteria:

$$\Big| - \frac{1}{n} \log_2 P_{X^n}(\boldsymbol{x}^n) - H(X) \Big| \leq \epsilon \tag{3.3}$$

Next, we will demonstrate how this typical set [34] concept can be exploited to compute atypical score for Markov chain.

3.3.2 Asymptotic Equipartition Property for Markov Chain

In this section, we will show how to compute the atypical score for a Markov chain, motivated by the assumption that sequential activities exhibit Markovian property. We aim to exploit the Asymptotic Equipartition Property (AEP) for Markov Chain in computer vision problem. This has been a well-established theorem [33, 3] applied to several other domains such as data compression, and data transmission. Fig. 3.3 shows an example of different activities in a video that are connected via a temporal link. We can assume this temporal ordering in terms of Markov chain, where current activity only depends on previous activity. Let us consider a stochastic process, where states can be denoted as $\{X_1, X_2, \dots, X_n\}$ and each state $X_i \in \mathcal{X}$. If a source $X_1, X_2, \dots X_n$, produces a sequence, we can characterize the distribution of a sequence as $p\{(X_1, \dots, X_n) = (x_1, \dots, x_n)\} = p(x_1, \dots x_n)$. Since we assume the temporal link as Markov chain, we can write the conditional independence as follows.

$$P(x_{n+1}|x_n, \dots, x_1) = p(x_{n+1}|x_n) \tag{3.4}$$

In Markov chain, one state moves successively to next state with a probability. Let us denote current state X_i which moves to next state X_j with probability p_{ij}. The probability p_{ij} is called transition probability. In activity recognition, we assume that the transition probability does not change over time. So, the Markov chain becomes time-invariant (stationary) where the conditional probability $p(x_{n+1}|x_n)$ does not rely on n. This can be written as

$$p_{X_1 \ldots X_n}(x_1, \ldots, x_n) = p_{X_{1+t} \ldots X_{n+t}}(x_1, \ldots, x_n). \tag{3.5}$$

Here, t denotes time shift. For stationary Markov chain, we can define a transition matrix T_s, where each entry represents the probability of jump from one state to another. The transitional matrix T_s can be written as

$$T_s = \begin{bmatrix} p_{11} & p_{12} & p_{13} & \cdot & p_{1n} \\ p_{21} & p_{22} & p_{23} & \cdot & p_{2n} \\ \multicolumn{5}{c}{\dotfill} \\ p_{n1} & p_{n2} & p_{n3} & \cdot & p_{nn} \end{bmatrix}$$

where, each $p_{ij} > 0$ and for all states X_i,

$$\begin{aligned} \sum_{m=1}^{n} p_{im} &= \sum_{m=1}^{n} p(x_m|x_i) \\ &= 1. \end{aligned} \tag{3.6}$$

Now, we can compute the probability of seeing a sequence, $p(X_1, X_2, \ldots, X_q)$ as

$$p(X_1, X_2, \ldots, X_q) = p(X_1)p(X_2|X_1)\ldots p(X_q|X_{q-1}) \tag{3.7}$$

Here, q is the number of elements in a sequence. If the Markov chain is stationary, then we can define a stationary distribution μ over all X_i. The stationary distribution can be computed as

$$\mu = \mu T_s \tag{3.8}$$

where, each element of μ would be $\mu_i = \sum_j^n \mu_j p_{ji}$, and $\sum_{i=1}^n \mu_i = 1$. If we transpose Eqn. 3.8, we obtain

$$\begin{aligned} (\mu T_s)^\intercal &= \mu^\intercal \\ T_s^\intercal \mu^\intercal &= \mu^\intercal \end{aligned} \tag{3.9}$$

Thus, stationary distribution can be obtained from the eigenvector of $T_s^\intercal$ with eigenvalue 1 by utilizing eigen value decomposition. If the transition matrix T_s is known, we can easily compute stationary distribution. It could be possible to have multiple eigenvectors associated to an eigenvalue of 1 where each eigenvector gives rise to an associated stationary distribution. In this case, the Markov chain becomes reducible, i.e. has multiple communicating classes [3].

Entropy Rate: The entropy rate of a stochastic process $\{X_1, X_2, \ldots, X_n\}$ can be written as

$$H(\mathcal{X}) = \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) \tag{3.10}$$

For stationary process, the entropy rate [93] becomes

$$\begin{aligned} H(\mathcal{X}) &= \lim_{n\to\infty} \frac{1}{n} H(X_1, X_2, \ldots, X_n) \\ &= \lim_{n\to\infty} \frac{1}{n} H(X_n|X_{n-1}, \ldots, X_1). \end{aligned} \tag{3.11}$$

$H(X_n|X_{n-1}, \ldots, X_1)$ is non-increasing as n increases and the limit must exist [93]. For Markov chain, the above equation 3.11 would be $H(\mathcal{X}) = \lim_{n\to\infty} H(X_n|X_{n-1}, \ldots, X_1) = \lim_{n\to\infty} H(X_n|X_{n-1})$. If $X_1 \sim \mu$, then the entropy rate is

$$H(\mathcal{X}) = -\sum_{ij} \mu_i p_{ij} \log p_{ij} \tag{3.12}$$

Using Eqns. 3.9 and 3.12, we can compute stationary distribution and entropy rate. From the Asymptotic Equipartition Property (AEP) theorem, the probability of a sequence (Eqn. 3.7) becomes

$$p(X_1, \ldots, X_q) \to 2^{-qH(\mathcal{X})}. \tag{3.13}$$

Now, we introduce a notation $\mathcal{E}$, which represents atypical score. $\mathcal{E}$ can be computed as $\mathcal{E} = -\frac{1}{q} \log p(X_1, \ldots, X_q) - H(\mathcal{X})$. Next, we will demonstrate how this atypical score can be utilized in a couple of applications- (a) sample selection and (b) anomaly detection.

3.3.3 Computation of Atypical Score in Video Applications

In activity classification, an activity generally shares the temporal relation with past activities. An activity might be strongly correlated with its previous activity, and it is also possible that two consecutive activities are uncorrelated. Thus, we consider that a temporal link is established if the time interval between current and previous activities is below a threshold τ, else it is possible that two consecutive activities are not temporally related. Fig. 3.3 shows an example of such scenario. From the figure, we can see that temporal link between last two activities is not established due to a long time interval. In this work, we assume that the current activity only depends on previous activity, thus $p(a_3|a_2, a_1) = p(a_3|a_2)$ as shown in Fig. 3.3.

As the activities form a sequence and generate Markov chain for a video clip, we can compute atypical score by computing entropy rate and the probability distribution of a sequence. For transition matrix, we simply count the frequency of an activity appearing given the previous activity. Consider, i^{th} row vector $\boldsymbol{r}^i$ of matrix T_s shown in Eqn. 3.6, which can be written as

$$\boldsymbol{r}^i = \frac{1}{\sum_{k=1}^{N_a} \phi_k^m} [\phi_1^i, \dots, \phi_n^i]. \tag{3.14}$$

Here, N_a represents the number of activity classes. ϕ_j^k implies the number of appearing activity class a_j with previous activity a_k. Thus, each (i,k)-th entry of $\boldsymbol{r}^i$ represents the transitional probability from a_i to a_k. After obtaining the transition matrix, we can easily compute stationary probability μ_a using Eqn. 3.9 by utilizing eigen value decomposition. Let us define the states $A_1, \dots, A_p$ and each state $A_i \in \mathcal{A}$. Each of these states can have outcome $a_1, \ldots, a_{N_a}$. So, we can compute the entropy rate as follows.

$$H(\mathcal{A}) = -\sum_{ij} \mu_{ai} \boldsymbol{r}_j^i \log \boldsymbol{r}_j^i \tag{3.15}$$

Given a video, set of activity sequences $A_1, \ldots, A_p$ are observed. p is the number of appearing objects in a video. The probability of a sequence would be $p(A_1, \ldots, A_p) = p(A_1)p(A_2|A_1) \ldots p(A_p|A_{p-1}) = p(\mathcal{A}^p)$. It can be calculated from μ_a and $\boldsymbol{r}_j^i$. We can compute the atypical score of the sequence as follows.

$$\mathcal{E} = -H(\mathcal{A}) - \frac{1}{P} \log_2 p(\mathcal{A}^P) \tag{3.16}$$

Now, in order to compute the atypical score for each of the samples, we remove a tuple from the sequence associated with activity sample a_t and observe the deviation of atypical score as similar to Eqn. 3.16. It might be possible that activity label at time t (a^t) is excluded as because it appears very far from activities before and after it. In an extreme case, we only compute the entropy of that activity sample, which will be discussed in Sec. 3.3.4. If we remove q^{th} sample from the sequence, then we compute the probability of a new sequence as $p(\mathcal{A}^{p'}) = p(A_1)p(A_2|A_1)...p(A_{q-1}|A_{q-2})p(A_{q+2}|A_{q+1})...p(A_P)$. Thus, $p(A_q|A_{q-1})$ and $p(A_{q+1}|A_q)$ are eliminated from the distribution function. The length of new sequence would be $P-2$. So, the atypical score of new sequence would be $\mathcal{E}_q$, which can be written as

$$\mathcal{E}_q = -H(\mathcal{A}) - \frac{1}{P-2} \log_2 P(\mathcal{A}^{P-2}) \tag{3.17}$$

We can now measure the deviation between $\mathcal{E}$ and $\mathcal{E}_q$.

$$\begin{aligned}
\tilde{\mathcal{E}}_q &= |\mathcal{E} - \mathcal{E}_q| \\
&= | -\frac{1}{P}\log_2 p(\mathcal{A}^P) + \frac{1}{P-2}\log_2 p(\mathcal{A}^{P-2})| \\
&= |\frac{2}{P(P-2)} \sum_{\substack{m=1 \\ m \neq \{q-1,q\}}}^{P} \log_2 p(A_{m+1}|A_m) - \\
&\quad \frac{1}{P}(\log_2 p(A_q|A_{q-1}) - \log_2 P(A_{q+1}|A_q)|
\end{aligned} \tag{3.18}$$

In case of first and last activity samples in a video, we only remove one element (either $p(A_1)$ or $p(A_p|A_{p-1})$). Finally, we compute an atypical score for each activity sample as, $\frac{\tilde{\mathcal{E}}_q}{N_t}$, where N_t denotes the number of tuple removed from the original sequence. Using the atypical scores, we can formulate our optimization problem to select the informative samples for manual labeling as discussed next.

Algorithm 2: Computation of Atypial Score and Uncertainty for Sample Selection

```
INPUTS. 1. Learned models from training data $\mathcal{L}$: Classification Model $\mathcal{M}$
and Transition Matrix $T_s$
        2. Unlabeled Video Clips: $\mathcal{U}$
OUTPUTS. The vectors for atypical Score $\mathcal{T}$ and entropy $\boldsymbol{h}$ for the
unlabeled data $\mathcal{U}$
Step 1: Compute Stationary Probability $\mu$ as shown in Eqn. 3.9 using $T_s$.
Step 2: Compute Entropy Rate $H(\mathcal{A})$ using $\mu$ and $T_s$ as in Eqn. 3.15.
Step 3: Obtain an activity sequence from the predicted labels provided by
  $\mathcal{M}$ for a video clip in $\mathcal{U}$.
Step 4: Compute the probability of sequence $p(\mathcal{A}^P)$ using $\mu$ and $T_s$ as in
  Eqn. 3.7
Step 5: Compute atypical score $\tilde{\mathcal{E}}_q$ using Eqn. 3.18 and entropy $h_q$
  associated with $q^{th}$ sample.
Step 6: Calculate vectors $\mathcal{T}$ and $\boldsymbol{h}$ as discussed in Sec. 3.3.4
```

3.3.4 Informative Sample Selection for an Activity Sequence

In this section, we will formulate an objective function from the atypical score of samples as discussed before. This objective function will be optimized to select the most informative samples. Let us consider that we have a batch of N unlabeled instances and we need to select the optimal instances for manual labeling. Let us define a vector $\boldsymbol{\mathcal{T}}_j = [\tilde{\mathcal{E}}_1 \quad \tilde{\mathcal{E}}_2 \quad \ldots]^T$, containing the atypical score of each sample of the j^{th} video depending on the recognition task (e.g., activity recognition) as in Eqn 3.18.

Consider a vector $\boldsymbol{\mathcal{T}}$ which represents the atypical scores of the samples for all videos. We can write $\boldsymbol{\mathcal{T}}$ in terms of $\boldsymbol{\mathcal{T}}_j$ as follows.

$$\boldsymbol{\mathcal{T}} = [\boldsymbol{\mathcal{T}}_1 \quad \boldsymbol{\mathcal{T}}_2 \quad \ldots]^T \tag{3.19}$$

We also incorporate the uncertainty of current baseline classifier on the unlabeled samples. We define a vector that denotes the entropy of all samples as $\boldsymbol{h} = [h_1 \quad h_2 \quad \ldots]^T$, where $h_j = \mathbb{E}[-\log_2 p_j]$, and p_j is the p.m.f. of prediction scores by the current baseline classifier on the j^{th} unlabeled sample. We aim to choose a subset of the samples which are informative based on the two criterion, namely atypical score and the entropy of each sample obtained from the classifier. We can write the optimization function in vector form as follows,

$$\begin{aligned} \boldsymbol{y}^* \quad &= \quad \arg\max_{\boldsymbol{y}} \; \boldsymbol{y}^T(\boldsymbol{h} + \lambda\boldsymbol{\mathcal{T}}) \\ s.t. \qquad & \boldsymbol{y} \in \{0,1\}^N \;, \quad \mathbf{y^T 1} \leq \eta \end{aligned} \tag{3.20}$$

Here, λ is a weighting factor. The term $\boldsymbol{y}^T\mathbf{1}$ represents the number of samples will be chosen which is bounded by η. Let us denote $\boldsymbol{f} = -(\boldsymbol{h} + \lambda\boldsymbol{\mathcal{T}})$. Maximization of the objective function in Eqn. 3.20 is the same as minimization of $\boldsymbol{y}^T\boldsymbol{f}$. It is a binary linear integer programming problem and can be solved by CPLEX [37]. Algorithm 2 shows the steps of our proposed method for selecting informative samples. Next we show how the sample selection strategy can be used for active learning and anomaly detection as two applications for the experiments.

```
Algorithm 3: Sample Selection for Active Learning with Continuous Data
    INPUTS. 1. Learned models at $\text{Batch}_{k-1}$ : Classification Model $\mathcal{M}_{k-1}$ and
    Transition Matrix $T_s^{k-1}$
            2. Unlabeled Video Clips at $\text{Batch}_K$: $\mathcal{U}_k$
    OUTPUTS. Learned Models after processing videos in $\text{Batch}_K$: $\mathcal{M}_k$ and $T_s^k$
    Initialize: $\mathcal{L} = \{L_0\}$ (Initial Set of Data)
    Step 1:  Calculate vectors $\mathcal{T}$ and $\boldsymbol{h}$ using Algorithm 2
    Step 2: Find optimal set of samples $\boldsymbol{y}_k^*$ using Eqn. 3.20 for Batch $k$
    Step 3: $\mathcal{L} = \mathcal{L} \cup \boldsymbol{y}_k^*$
    Step 4: Update models $\mathcal{M}_{k-1}$ and $T_s^{k-1}$ with $\mathcal{L}$.
```

Active Learning

The sample selection strategy discussed above can be used in an active learning framework to update a classification model online. The adaptability of recognition models to the continuous data stream becomes important for long-term performance. Given a set of data at particular time, the proposed sample selection approach can be utilized to select the most informative samples in order to update the model. After obtaining a set of samples $\boldsymbol{y}^*$ from Eqn. 3.20, we can ask a human to label these samples. With newly generated labeled data, the classification model $\mathcal{M}$, and the temporal relationship model need to be adjusted.

Update $\mathcal{M}$. For classification task, we use softmax classifier to predict the labels. If the feature vector is $\mathcal{F}_k$ for k^{th} sample, then predicted probability for the j^{th} class can be written as, $P(l = j|\mathcal{F}_k) = \frac{e^{\mathcal{F}_k^T w_j}}{\sum_{k=1}^{K} e^{\mathcal{F}_k^T w_k}}$. Here, K is the number of classes, w_j represents the weights corresponding to class j. We optimize the cross entropy loss function to estimate the parameters as presented in [38]. For the current batch, we update the parameters with the newly labeled data samples.

Update Temporal Relationship Model. Let us consider a matrix Φ that represents the temporal statistics between activities. Φ will be updated based on the newly acquired labels. The updated statistics can be written as, $\Phi' \leftarrow \Phi + \tilde{\Phi}$, where $\tilde{\Phi}(.)$ represents the statistics with the newly labeled samples and Φ' is the updated statistics. With updated Φ, transition matrix T_s is modified.

```
Algorithm 4: Algorithm of Proposed Method for Anomaly Detection
    INPUTS. 1. Learned models with normal activities : Classification Model
    $\mathcal{M}$ and Transition Matrix $T_s$
            2. Test Video Clip $\mathcal{V}_t$.
    OUTPUTS. Set of binary labels $\mathcal{C}$ for anomaly and normal activities for $\mathcal{V}_t$.
    Step 1:  Compute atypical score $\tilde{\mathcal{E}}_j$ using Eqn. 3.18 and entropy $h_j$
      associated with $j^{th}$ sample using Algorithm 2.
    Step 2: Calculate irregularity score $\mathcal{D}_j$ using Eqn. 3.21.
    Step 3: Assign class labels $\mathcal{C}$ for all the activities in $\mathcal{V}_t$ based on threshold $\tau$
      as discussed in  3.3.4.
```

Anomaly Detection

In anomaly detection, we consider an activity as abnormal if it is an outlier with respect to the learned model. Thus, any prior information on anomalous activity at training time is unknown. We learn the recognition model $\mathcal{M}$ from the regular activities. The temporal relationship between the activity samples is also exploited during the learning process. We compute transition matrix T_s from this temporal relations. We calculate stationary distribution μ followed by entropy rate $H(\mathcal{X})$ using Eqns. 3.9 and 3.12.

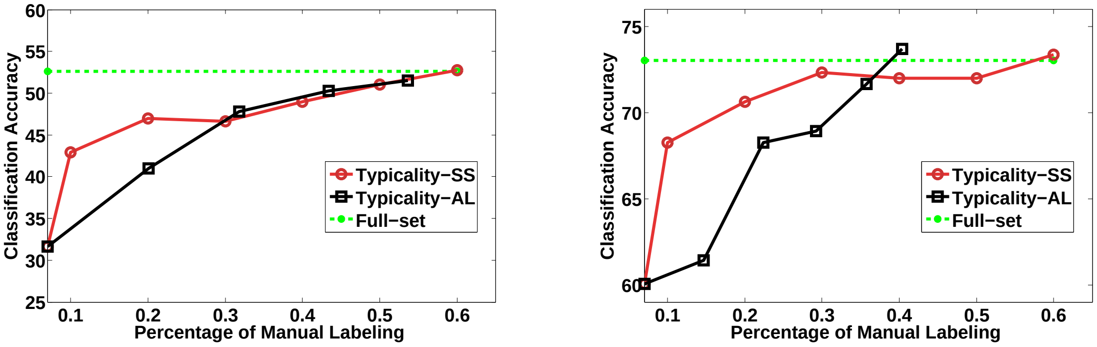
Given a test video, $\mathcal{M}$ predicts the activity labels, from which a sequence is formed. We can compute atypical score $\tilde{\mathcal{E}}_j$ associated with j^{th} sample as discussed in Sec. 3.3.3 using Eqn. 3.18. We also compute the entropy $h_j = \mathbb{E}[-\log_2 p_j]$ from the distribution of confidence score provided by $\mathcal{M}$. We can now define irregularity score $\mathcal{D}_j$ which can be written as

$$\mathcal{D}_j = \tilde{\mathcal{E}}_j + \beta h_j. \tag{3.21}$$

β represents weighting factor. We also consider entropy along with the atypical score in order find an anomaly. Given an anomaly class, entropy should be high as it exhibits high uncertainty. All the steps are demonstrated in Algorithm **??**. If $\mathcal{D}_j$ is larger than a threshold τ then it is considered as an abnormal class, or normal otherwise. The class of a sample C_j can be determined as follows.

$$C_j = \begin{cases} 1, & \text{if } \mathcal{D}_j > \tau \\ 0, & \text{otherwise} \end{cases}$$

Here, 1 represents abnormal activity and 0 denotes normal class. Next, we will demonstrate the experimental analysis of our proposed approach to sample selection and anomaly detection.



(a) MPII-Cooking Dataset

(b) VIRAT Dataset

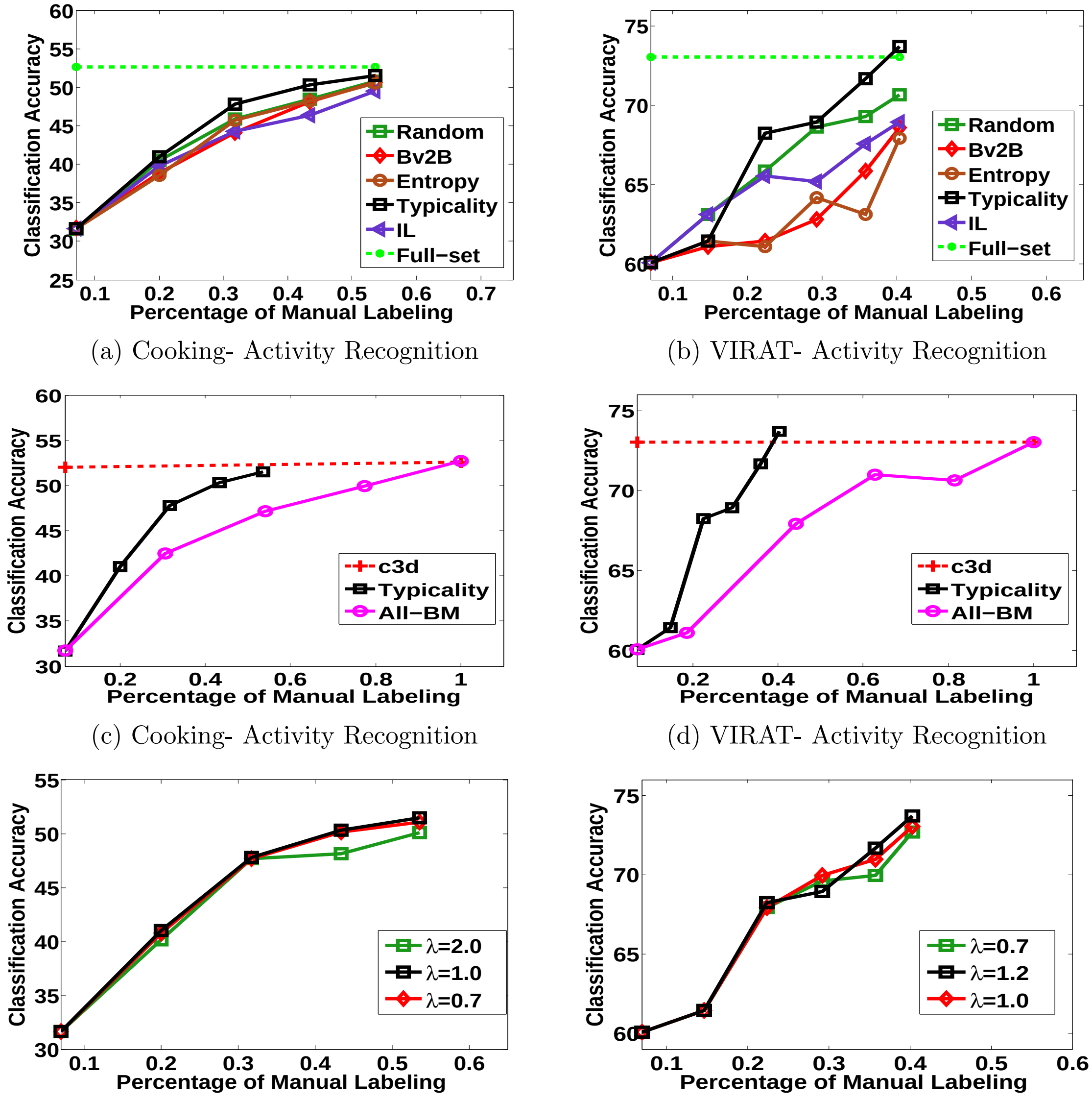
Figure 3.4: This figure illustrates the recognition performance of the proposed method for the tasks of informative sample selection and active learning, on (a) MPII-Cooking, and (b) VIRAT datasets.

3.4 Experiments

In this section, we evaluate our proposed method on two distinct applications such as informative sample selection for recognition model, and anomaly detection, for activity recognition task. We also compare our methods with state-of-the-art approaches on two challenging datasets.

Datasets. We demonstrate the performance of our proposed method on two video datasets. We evaluate our results on VIRAT [116] and MPII-Cooking [128] datasets for activity classification task. VIRAT is a video dataset which provides temporal relations between different activity samples. This dataset has 329 video clips consisting of 11 different activities [116]. MPII-Cooking dataset presents 65 cooking activities, e.g., *cut slices, pour, or spice* [128]. It has 44 videos in total. Since videos are usually long, we follow sliding window approach for cropping short video clips in order to create more video instances.

Feature Extraction. For activity recognition model, we adopt the classification model described in [142]. We utilize the final layer of 3d convolutional neural network



(a) Cooking- Activity Recognition

(b) VIRAT- Activity Recognition

(c) Cooking- Activity Recognition

(d) VIRAT- Activity Recognition

(e) Cooking- Activity Recognition

(f) VIRAT- Activity Recognition

Figure 3.5: The figure presents the performance of proposed active learning method for activity recognition task on two datasets - MPII-Cooking [128] (first column) and VIRAT [116] (second column) datasets. Plots (a,b) present the comparison against other state-of-the-art sample selection methods. Plots (c,d) demonstrate comparison with BM-All method. Plots (e,f) demonstrate the sensitivity analysis of our framework. Best viewable in color.

to extract features. Finally, we have 4096 dimensional c3d [142] feature for an activity sample (small clip of a video). These features are used to train softmax classifier for activity recognition.

Evaluation Criterion. In order to evaluate active learning (AL) methods, we generate a plot of recognition accuracy vs percentage of manual labeling. We aim to achieve the same performance with less manual labeling effort. We utilize percentile (%) accuracy for activity recognition. For anomaly detection, we use ROC curve which measures the performance of binary classification task with varying threshold on prediction score. Finally, we calculate the area under the curve (AUC) to assess the performance. The value of AUC generally lies in between 0 and 1. We aim to achieve higher AUC value.

Experimental Setup. Our goal is to demonstrate two applications- (a) informative sample selection, and (b) anomaly detection, using proposed method discussed in Sec. 3.3. In order to choose the most informative samples, we consider two scenarios- (1) sample selection from fixed data, (2) batch-mode active (online) learning. In first scenario, we fix the percentage of manual labeling from the whole training set and measure the performance on test set. In this setting, proposed framework inspects all the samples while selecting the informative samples. We learn the initial model from very few samples which are excluded from the training set. In batch-mode active learning, we consider same experimental setting as [10], where data samples (videos) are continuously coming in batches. We first divide the dataset into training and testing set. We create 5/6 batches from the training set. We evaluate the recognition performance on the test set after processing of each batch. Initial models (classification and temporal relations) are learned from the first batch of data.

Typically, the first batch is smaller than other batches. Next, we apply sample selection strategy on next batches to choose the most informative samples. From the newly learned samples, models are updated. We also incorporate incremental learning to update the model as new classes can come in new batches. For anomaly detection, we learn the recognition and temporal relations from the normal activities. Now, given a test video, we compute irregularity score as discussed in Sec. 3.3.4, on which we determine whether an activity is anomalous or not.

State-of-the-art and Baseline Methods: In the experiment, we compare against different existing approaches and some baseline methods. These methods are as follows.

◇ **Typicality-SS:** Proposed approach applied to informative subset selection.

◇ **Typicality-AL:** Typicality based sample selection strategy for active (or online) learning task.

◇ **Bv2B:** Best vs Second Best active learning strategy [83].

◇ **IL:** Incremental learning approach presented in [59].

◇ **Full-set:** Entire training is used to obtain the accuracy from baseline classifiers.

◇ **BM-All:** All the samples in current batch are considered.

The baseline methods mentioned above are implemented on our training and testing set for fair comparison.

3.4.1 Informative Subset Selection from Fixed Data

In order to evaluate the performance of our sample selection strategy discussed in Sec. 3.3.4, we vary the percentage of manual labeling from the training set, and measure the performance on test set. We keep the initial set fixed which is learned from very few samples. In this experimental setup, whole training set is observed in sample selection process. On the contrary, data are coming into batches for active learning. In our experiment, we choose 10% to 60% with 10% increment as the percentage of manual labeling, and compute the recognition accuracy for activity recognition. Fig. 3.4 illustrates the performance of our proposed method on sample selection. In this figure, we plot the classification accuracy of our proposed method with varying the percentage of manual labeling on two applications-sample selection and active learning. Typicality-SS and Typicality-AL represent the proposed approach for sample selection and active learning respectively. From this figure, we observe that the recognition performance of typicality-SS outperforms typicality-AL as the percentage of manual labeling decreases. The underlying reason is that typicality-SS considers the whole training set during the sample selection process unlike typicality-AL where active learner only utilize small portion of full dataset.

3.4.2 Performance of Batch-Mode Active Learning

We perform a various set of experiments to evaluate our proposed framework for online learning. We analyze the following experiments: 1. Comparison with existing active learning approaches, 2. Comparison against baseline methods, 3. Sensitivity analysis of the parameters, and 4. Time complexity of the proposed method, and 5. Performance with varying sequence length.

Comparison With Other Active Learning Methods

We compare our active learning (AL) approach with other state-of-the-art methods and baseline approaches as mentioned above. Figs. 3.5(a,b) show the recognition performance with respect to the percentage of manual labeling. We observe the performance on test set with updated recognition model after processing each batch of data. The straight line presented in the figures implies recognition accuracy of the model with 100% manual labeling (whole training set). We compare with some of the existing AL approaches such as Bv2B [83], random sample selection, Entropy [41] and IL [59]. For comparison, we first run our AL method to obtain the number of samples, which will be manually labeled. Then, we fix the number of samples for each batch and obtain the accuracy for other AL methods. In other words, different AL methods select the different subset of samples from each batch, where the size of subsets would be same. The performance will vary due to the selection of different subsets. For a fair comparison, we also keep same features and baseline classifiers for all the methods. From Figs. 3.5(a,b), we can see that the proposed framework *outperforms other AL methods to reduce the manual labeling cost by a large margin* in activity classification. Our method requires only **54%**, and **40%** of manual labeling to achieve the optimal recognition performance on VIRAT [116] and MPII-Cooking [128] datasets respectively as shown in Figs. 3.5(a,b). From Figs. 3.5(a,b), we can also see the performance gap between our method and other approaches. In MPII-Cooking [128] dataset, our approach outperforms Bv2B [83], random sample selection, Entropy [41] and IL [59] by **0.89%**, **0.67%**, **0.97%** and **2.00%** respectively with 54% manual labeling. Similarly, for VIRAT [116] dataset, proposed method surpasses Bv2B [83], random sample selection, Entropy [41] and IL [59] by **5.12%, 3.07%, 5.80%** and **4.78%** respectively with 40% annotation effort as shown in Fig. 3.5.

Comparison Against Other Baseline Methods

To evaluate proposed approach, we compare against BM-All method for activity classification. BM-ALL represents all the samples in a current batch, thus for N_b batches we have N_b accuracy values. Figs. 3.5(c,d) show the plots of our proposed model and BM-All method. BM-ALL helps us to understand the effectiveness of proposed method in selecting the most informative samples. We aim to achieve similar performance with BM-All with less manual labeling effort. From the comparison of BM-ALL and proposed method, we can observe that a good recognition model can be learned from a small set of informative samples. Figs. 3.5(c,d) demonstrate that the proposed framework achieves similar or better performance with fewer informative samples when compared to BM-All method. In Fig. 3.5(d), we can also see that the proposed method outperforms the model with 100% labeling (red straight line). This also attests that informative (quality) data is often more useful than simply more data (quantity).

Sensitivity Analysis of the Parameters.

In the proposed framework, we use the parameter λ as discussed in Sec. 3.3.4. In order to understand the efficacy of typicality, we show different plots with varying λ in Figs. 3.5(e,f). We set the values of λ ranging from 0.7 to 2.0. We empirically choose these values to observe the change in plots. Figs. 3.5(e,f) illustrate the variation of performance due to change in hyperparameter λ. With high value of λ, we put more weight on atypical

Method	Cooking [128] Time(s)	VIRAT [116] Time(s)
Proposed Method	$62.75s$	$69.84s$
BM-All Method	$2498.32s$	$3281.08s$

Table 3.1: Analysis of computation time on MPII-Cooking [128] and VIRAT [116] datasets. We can see from the table that our approach reduces computation time during training of recognition model.

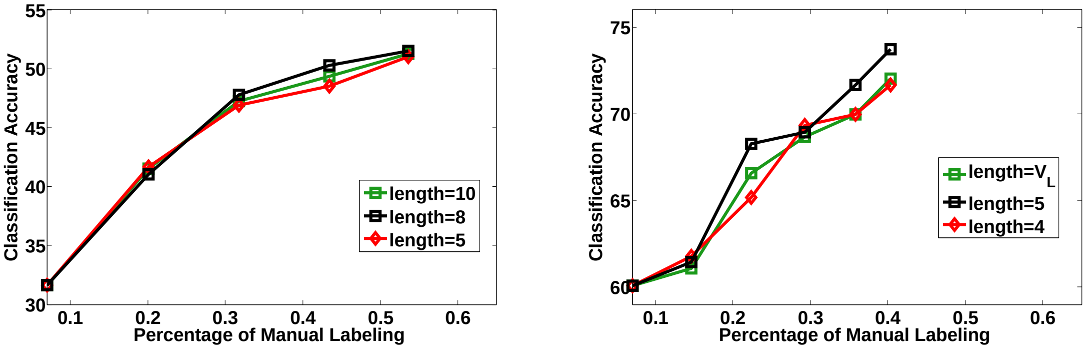
score (Sec. 3.3.4). From figures, we can see that the performance degrades with the smaller value of λ.

Time Complexity.

The proposed method also reduces computation time to adapt the recognition model. Table. 3.1 shows the computational time on MPII-Cooking [128] and VIRAT [116] datasets. We compute the time to query the samples, and time to train recognition models for a dataset. We also compute the time to train a recognition model with all the samples in a batch (BM-All method). As we can see that total time to train activity model with all the samples is $3281.08s$ for MPII-Cooking [128], and $69.84s$ for VIRAT [116] dataset. On the other hand, the total time for querying and training with samples selected by our approach is $2498.32s$, and $62.75s$ for MPII-Cooking [128] and VIRAT [116] datasets respectively. From the Table. 3.1, we can see that the proposed AL method helps to save a significant amount of computational time, especially in a big dataset.

Performance with Varying Sequence Length

We also set up an experiment in order to observe the effect of varying sequence length on recognition performance for active learning. We consider both MPII-Cooking and



(a) MPII-Cooking Dataset

(b) VIRAT Dataset

Figure 3.6: This figure illustrates the recognition performance with varying sequence length on (a) MPII-Cooking, and (b) VIRAT datasets.

VIRAT datasets to run this experiment. Sequence length represents the number of activities in a video clip. In order to prepare the data, we extract video clip with varying sequence length from the original video by following sliding window. For MPII-Cooking dataset, we vary the sequence length to $10, 8$ and 5. We consider the whole video (length=V_L) and two different lengths (4 and 5) for VIRAT dataset. The performance of proposed method with varying sequence length on MPII-Cooking and VIRAT datasets is demonstrated in Fig. 3.6(a) and 3.6(b) respectively. From the figures, higher performance is observed for longer sequence.

3.4.3 Anomaly Detection

In this section, we will show how atypical score (discussed in Sec. 3.3.3) can be utilized to detect anomaly activities. In order to evaluate the performance in anomaly detection, one class is randomly chosen as abnormal, and rest are considered as normal activities. We perform cross-validation to assess the performance. We learn the temporal

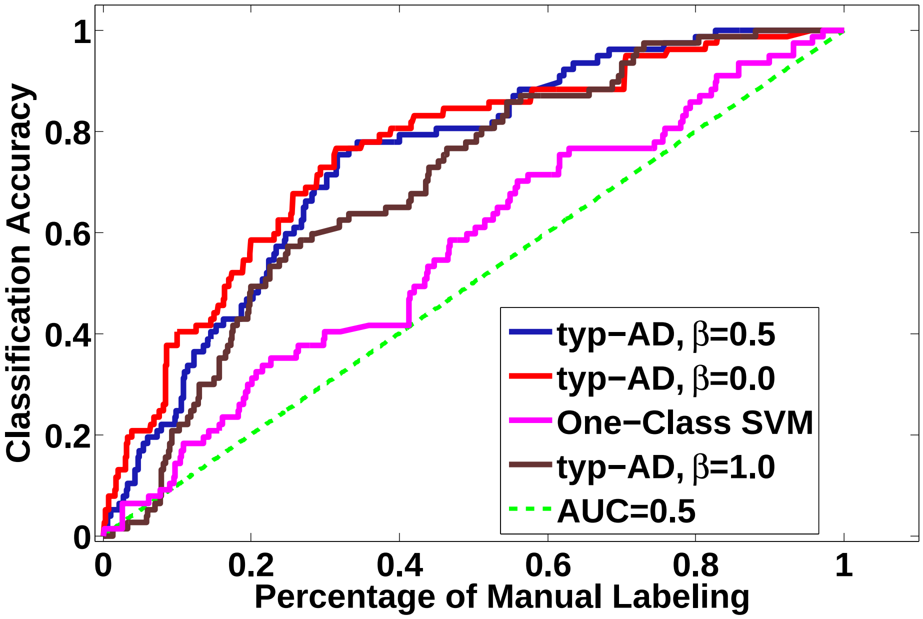


Figure 3.7: The figure shows ROC plots for anomaly detection on VIRAT [116] dataset. Different colors represent baseline methods. Best viewable in color.

relations from normal activity classes. To train the recognition model, we train multi-class softmax classifier with normal activities. We exclude the abnormal class during the learning process. Given a test video, recognition model provides a probability distribution over the classes. We compute entropy from this distribution. If the activity class belongs to normal activities, recognition model shows low uncertainty. For abnormal class, the uncertainty goes high. We also calculate the atypical score for the activity samples in test video. After computing the uncertainty and atypical score, we calculate the irregularity score using Eqn. 3.21. Based on this irregularity score, we determine whether an activity is abnormal or not.

In order to evaluate our framework, we plot ROC curve by varying the threshold on irregularity. Fig. 3.7 shows ROC plots for different methods. We consider One-class SVM

Method	AUC Score
typ-AD with $\beta = 1.0$	0.70
typ-AD with $\beta = 0.5$	0.75
One-Class SVM [137]	0.57
Context-Aware Model [166]	0.685
typ-AD with $\beta = 0$	0.76

Table 3.2: Analysis of computation time on MPII-Cooking [128] and VIRAT [116] datasets. We can see from the table that our approach reduces computation time during training of recognition model.

[137] as the baseline method to detect anomalous activity. For convenience, we refer our proposed method as 'typ-AD' (i.e, typicality for Anomaly Detection) in Fig. 3.7. In typ-AD, we change the value of β as discussed in Sec. 3.3.4 ranging from 0 to 1.0 to observe the effect on uncertainty score. From the figure, we can see that the proposed method with only atypical score ($\beta = 0$) outperforms all other methods. As the value of β increases, we put more weights on entropy h_j as shown in Eqn. 3.21. We can see from the Fig. 3.7 that the performance of anomaly detection is improved as the value of β decreases.

We also provide area under the curve (AUC), which is computed from ROC curves as shown in Fig. 3.7 to measure the performance of anomaly detection. Table 3.2 illustrates the value of AUC for different methods. As we change the value of β, we observe different performance. With values of $\beta = 0.5$ and $\beta = 1.0$, we obtain AUC of 0.75 and 0.70 respectively. We observe the best performance with $\beta = 0$, which achieves 0.76 in AUC value. We also compare against baseline (one-class SVM [137]) and other existing method (Context-Aware Model [166]). The AUC values for One-class SVM [137] and Context-Aware Model [166] are 0.57 and 0.685 respectively.

3.5 Conclusions

In this paper, we presented a subset selection method by exploiting information-theoretic ‘typical set’ to adaptively learn the recognition models. We show that ‘typical set’ is a powerful tool which has been successfully used in data processing and can also be utilized in informative subset selection problem for visual recognition tasks. Our method is applied to various applications including sample selection and anomaly detection. The notion of typicality is used for a sequence of activities that can be represented as a Markov chain. Our approach significantly reduces the human load in labeling samples for visual recognition tasks. We demonstrate that our method achieves better or similar performance with only a small subset of the full training set compared with a model using full training set. Our model also shows good performance in anomaly detection in a video. As a future direction, we will study how typicality can be utilized to transfer knowledge from one domain where data is available to another where there is limited labeled data.

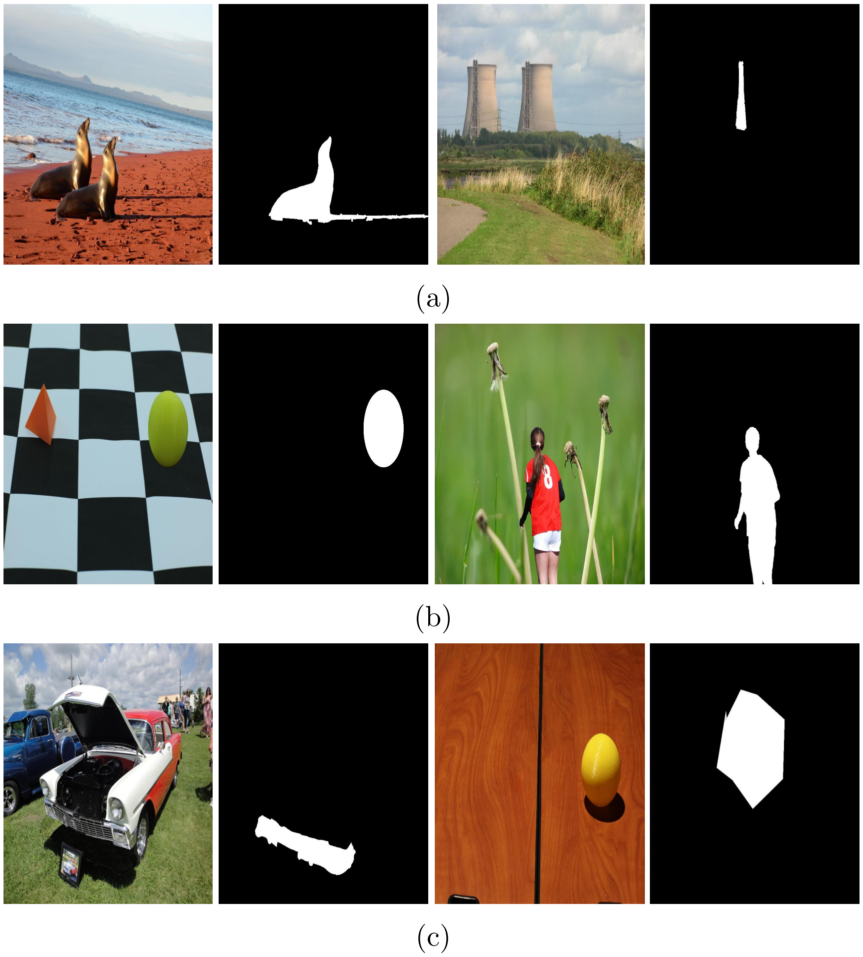
Chapter 4

Deep Learning Architecture for Detection of Image Forgeries

4.1 Introduction

The detection of image forgery has become very difficult as manipulated images are often visually indistinguishable from real images. With the advent of high-tech image editing tools, an image can be manipulated in many ways. The types of image manipulation can broadly be classified into two categories: (1) content-preserving, and (2) content-changing [68]. The first type of manipulation (e.g., compression, blur and contrast enhancement) occurs mainly due to post-processing, and they are considered as less harmful since they do not change any semantic content. The latter type (e.g., copy-move, splicing, and object removal) reshapes image content arbitrarily and alters the semantic meaning significantly [68]. The content-changing manipulations can convey false or misleading information. As the number of tampered images grows at an enormous rate, it becomes crucial to detect the manipulated images to prevent viewers from being presented with misleading information. Recently, the detection of content-changing manipulation from an image or a video has become an area of growing interest in diverse scientific and security/surveillance applications. In this chapter, we present a novel framework to localize manipulated regions at pixel level for content-changing manipulation.

Over the past decades, there have been lot of works to classify image manipulation, i.e., whether an image is tampered or not [110, 17, 79, 67, 123, 130, 50]. However, only few works [11, 18] attempt to localize image manipulation at pixel level. Some recent works [21, 44, 95] address the localization problem by classifying patches as manipulated. The localization of image tampering is a very challenging task as well-manipulated images do not leave any visual clues, as shown by the following examples. Some of the content-changing manipulation techniques are removing objects from an image, copy-clone, and splicing objects into image. Fig. 4.1 shows some examples of different techniques to tamper an image that changes the semantic meaning. In Fig. 4.1(a), copy-move manipulation is illustrated where one object is copied to another region of the same image leading to two similar objects, one originally present, and another manipulated. However, only the latter needs to be identified. Fig. 4.1(b) illustrates object splicing manipulation, where an object from a donor image has been spliced into other image. As another example, if an object is removed as shown in Fig. 4.1(c), the region may visually blend into the background, but needs to be identified as manipulated.



(a)

(b)

(c)

Figure 4.1: The figure demonstrates some examples of content-changing manipulations. (a), (b), (c) illustrate copy-clone, splicing and object removal techniques to manipulate an image. First and third columns are tampered images and corresponding ground-truth masks are shown in second and fourth columns.

Most of the state-of-the-art image tamper classification approaches utilize the frequency domain characteristics and/or statistical properties of an image [78, 98, 102, 149]. The analysis of artifacts by multiple JPEG compressions is also utilized in [25, 149] to detect manipulated images, which are applicable only to the JPEG formats. In [113, 112], noise has been added to the JPEG compressed image in order to improve the performance of resampling detection. In computer vision, deep learning has shown promising performance in different visual recognition tasks such as object detection [52], scene classification [165], and semantic segmentation [96]. Some recent deep learning based methods such as stacked auto-encoders (SAE) [163] and convolutional neural networks (CNN) [127, 12, 26] have also been applied to detect/classify image manipulations. In media forensics, most of the existing forgery detection approaches focus on identifying a specific tampering method, such as copy-move [23, 61, 80], splicing [105], etc. Thus, one approach might not do well on other types of tampering. Moreover, it seems unrealistic to assume that the type of manipulation will be known beforehand. Our recent paper [11], upon which this particular work builds, presents a general detection framework for different content-changing manipulations.

Unlike semantic object segmentation where all meaningful regions (objects) are segmented, the localization of image manipulation focuses only the possible tampered region which makes the problem even more challenging. In computer vision, recent advances in semantic segmentation methods [164, 96, 8] are based on convolutional neural networks (CNN). In CNN, the hierarchical features extracted at different levels help understand the content of the objects or shape of a region. In object detection [52] and segmentation [96, 8], CNN based architectures demonstrate very promising performance in understanding visual concepts by analyzing the content of different regions. In contrast to semantic segmentation, manipulated regions could be removed objects, or copied objects from other parts of the image. Well-manipulated images usually show strong resemblance between fake and genuine objects/regions (i.e. content is similar) [127]. Even though CNN generates spatial maps for different regions of an image, it can not generalize some other artifacts created by different manipulation techniques. Thus, the localization of manipulated regions with only CNN based architecture may not be the best strategy. In our earlier work [11], we compared with some recent semantic segmentation approaches [164, 96] that do not perform well for copy-clone and object removal type of manipulations.

Image tampering creates some artifacts, e.g., resampling, compression, shearing, which are better captured by resampling features [131, 21, 49]. In [21], resampling features are utilized as an important signature to detect manipulation. The authors trained six classifiers to detect six different types of resampling (e.g., JPEG quality thresholded above or below 85, upsampling, downsampling, rotation clockwise, rotation counterclockwise, and shearing (in an affine transformation matrix). Resampling introduces periodic correlations among pixels due to interpolation. As convolutional neural networks exhibit robust translational invariance to generate spatial maps for the different regions of the image, and certain artifacts are well-captured in resampling features, both can be exploited in order to localize manipulated regions.

Towards the goal of localizing manipulated regions in an image, *we present a unified framework that exploits resampling features, LSTM network, and encoder-decoder architectures in order to learn the pixel level localization of manipulated image regions.* We

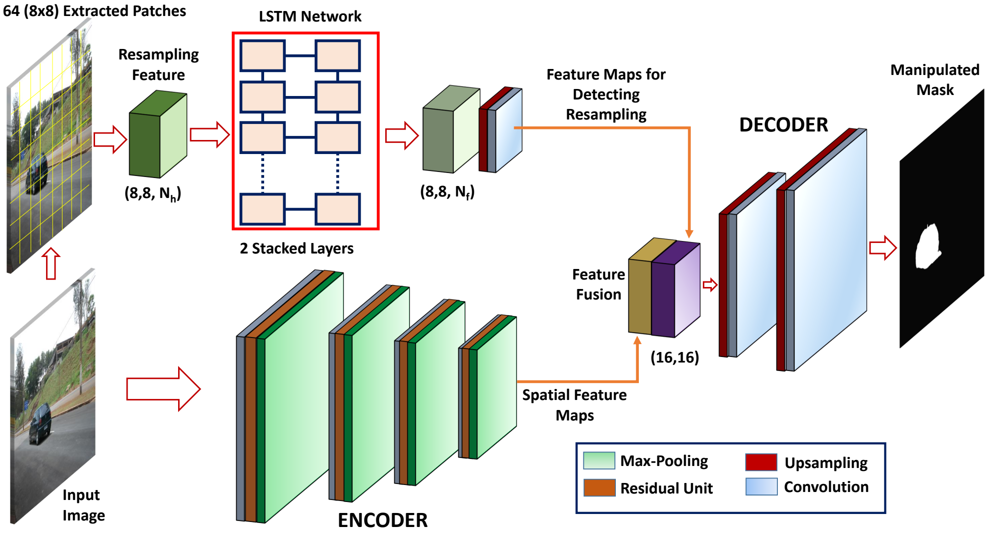


Figure 4.2: Overview of proposed framework for localization of manipulated image regions.

train this network end-to-end using back-propagation algorithm. As deep networks are data hungry, we create lots of synthesized images to learn the base model. The proposed model shows promising results in localizing manipulated regions at the pixel level, which is demonstrated on different challenging datasets.

4.1.1 Framework Overview

Given an image, our goal is to localize the manipulated regions at a pixel level. The proposed framework is shown in Fig. 4.2. Our network can be divided into three parts- (1) LSTM network with resampling features, and (2) convolutional encoder, and (3) decoder network.

For the first part, we divide image into patches. For each patch, resampling features [21] have been extracted. With extracted resampling features, we use Hilbert curve (discussed in Sec. 4.3.1) to determine the ordering of the patches to feed into LSTM cells. We allow

LSTM cells to learn the transition between manipulated and non-manipulated blocks in the frequency domain. Finally, feature maps are generated from the LSTM cell output, which will be combined with the feature maps from the encoder. An encoder consists of residual block, batch normalization and activation function. At each residual block, two convolutions are performed with shortcut connection. After each residual unit, max-pooling operation is performed which gives translation invariance.

Our next step is to design a decoder that can provide finer representation of different regions in a mask. We combine both spatial features from encoder and output feature from LSTM to understand the nature of manipulation. Then, these features are taken as input to the decoder. Each decoder follows basic operations like upsampling, convolution, batch normalization and activating feature maps (using activation function). The decoders help learn the finer details of the manipulated and non-manipulated classes. Finally, a softmax layer is used to predict manipulated pixels against non-manipulated ones. With the ground-truth mask of manipulated regions we perform end-to-end training to classify each pixel. We compute cross entropy loss, which is then minimized by utilizing back-propagation algorithm. After optimization, we find the optimal set of parameters for the network, that will be used to predict manipulated regions given a test set.

4.1.2 Main Contributions

Our *main contributions* are as follows.

• In this work, we propose a novel localization framework that exploits both frequency domain features and spatial context in order to localize manipulated image regions, which makes our work significantly different than other state-of-the-art methods.

- Unlike most of the existing works where patches are used as input, we consider image as input so that we can utilize global context. Our framework is able to localize manipulated region with high confidence, which is shown on two datasets.

- We present a new dataset for image tamper localization task that includes a large number of images with ground-truth binary mask. This dataset is bigger than any of the publicly available datasets such as IEEE Forensics [1] and COVERAGE [150]. It will also help train deeper networks for image tamper classification or localization tasks.

This work builds upon our earlier paper [11], but with significant differences. First, we propose new architecture where spatial features are learned by using an encoder network, and frequency domain features are exploited to observe the transition between patches by utilizing an LSTM network. Finally, a decoder network decodes the multi-modal feature space to localize the manipulated regions. Second, we consider image as input instead of patches that allows the network to learn larger context, i.e., intra-patch and inter-patch correlation. Third, unlike [11], we utilize resampling features in our network that captures characteristics of different artifacts due to image transformation such as upsampling, downsampling, rotation and shearing.

4.2 Related Work

In media forensics, there have been lot of efforts to detect different types of manipulations such as resampling, JPEG artifacts, and content-changing manipulations. In this section, we will briefly discuss some of the existing works for detecting image forgeries.

In last few years, several methods have been proposed to detect resampling in digital images [131, 111, 49]. Most of the approaches exploit linear or cubic interpolation. In [131], periodic properties of interpolation by the second-derivative of the transformed image have been utilized for detecting image manipulation. In [111], an approach was presented to identify resampling on JPEG compressed images where noise was added before passing the image through the resampling detector; it was shown that adding noise aids in detecting resampling. In [48, 49], a feature was generated from the normalized energy density and then SVM was used to robustly detect resampled images. Some recent approaches [55, 76] have been proposed to reduce JPEG artifacts produced by compression techniques. In [6, 143], feature based methods have been presented in order to detect manipulation in an image.

Many methods have been proposed to detect seam carving [135, 51, 94] and inpainting based object removal [152, 25, 88]. Several approaches exploit JPEG blocking artifacts to detect tampered regions [91, 45, 99, 16, 17]. Some recent works [80, 71, 67, 4] focus on identifying and localizing copy-move manipulation. In [80], the authors used an interesting segmentation based approach to detect copy-move forgeries. They first divided an image into semantically independent patches and then performed keypoint matching among these patches. In [36], a patch match algorithm was used to efficiently compute an approximate nearest neighbor field over an image. They further used invariant features such as Circular Harmonic transforms and showed robustness over duplicated blocks that have undergone geometrical transformations. In [105], an image splicing technique was presented using visual artifacts. In [109], the steerable pyramid transform (SPT) and the local binary pattern (LBP) were utilized to detect image forgeries. The paper [57] highlights the recent advances in image manipulation and also discusses the process of restoring missing or damaged areas in an image. In [7], a review on different image forgery detection techniques is presented.

Recently, there has been a growing interest to detect image manipulations by applying different computer vision and machine learning algorithms. In semantic segmentation, many deep learning architectures [96, 164, 8] have been proposed, which surpass previous state-of-the-art approaches by a large margin in terms of accuracy. Most of the deep networks [96, 8] are based on Convolutional Neural Networks (CNNs), where hierarchical features are exploited at different layers in order to learn the spatial map for semantic region. In [96], a classification-purposed CNN is transformed into fully convolutional one by replacing fully connected layers to produce spatial heatmaps. Finally, a deconvolution layer is used to upsample the heatmaps to generate dense per-pixel labeling. SegNet [8] designs a decoder to efficiently learn the low-resolution heatmaps for pixel-wise predictions for segmentation. In [73, 27], the fully connected pairwise CRF is utilized as a post-processing step to refine the segmentation result. In [120], skip connection is exploited to perform late fusion of feature maps for making independent predictions for each layer and merging the results. In ReSeg [145], Gated Recurrent Units (GRUs) and upsampling have been used to obtain the segmentation mask.

Recent efforts, including [12, 13, 127, 21, 107] in the manipulation detection task, exploit deep learning based models. These tasks include detection of generic manipulations [12, 13], resampling [14], splicing [127], and bootleg [20]. In [124], the authors propose Gaussian-Neuron CNN (GNCNN) for steganalysis. A deep learning approach to identify facial retouching was proposed in [15]. In [163], image region forgery detection has been performed using stacked auto-encoder model. In [12], a new form of convolutional layer is proposed to learn the manipulated features from an image. In computer vision, deep learning has led to significant performance gain in different visual recognition tasks such as image classification [165], and semantic segmentation [96]. The deep networks extract hierarchical features to represent the visual concept, which is useful in object segmentation. Most of the architectures are based on Convolutional Neural Network (CNN), which provides spatial maps relevant to manipulated regions. However, we can also exploit resampling features that distinguish other artifacts. Since, both spatial context and resampling are important attributes to localize manipulated regions from image, we present an unique network that exploits both of the features.

4.3 Network Architecture Overview

Our main goal of this work is to localize image manipulations at pixel level. Fig. 4.2 shows our overall framework. The whole network can be divided into three parts - (1) LSTM network with resampling features, and (2) Encoder, and (3) Decoder network. Convolutional neural network (CNN) architectures extract meaningful spatial features for object segmentation, which could also be useful to localize manipulated objects. Even though spatial feature maps are crucial to classify each pixel, solely using CNNs in the image domain does not usually perform well in identifying image manipulations. It is simply because there are certain manipulations like upsampling, downsampling, compression, which are well-captured in the frequency domain. Thus, we use resampling features from the extracted

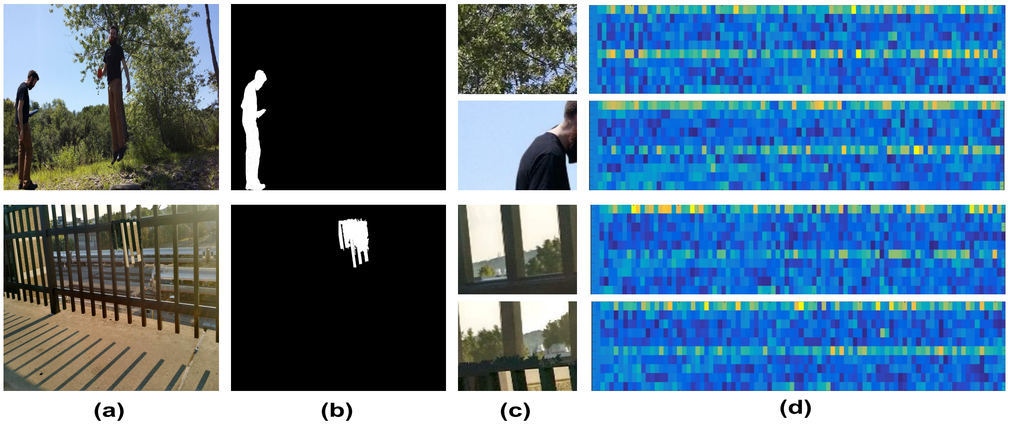


Figure 4.3: The figure shows example of manipulated images from the NIST dataset[2] column (a). Column (b) shows the corresponding ground-truth masks for the manipulated images in column (a). Columns (c) shows two patches extracted from the corresponding image, with the top containing no manipulation and the bottom one containing some manipulations. Columns (d) shows the radon transform corresponding to each of the extracted patches with each row corresponding to one of the 10 extracted angles.

patches of an image. These resampling features are considered as input to the LSTM network which learns the correlation between different patches. An encoder architecture is also utilized to understand the spatial location of manipulated region. Before decoder network, we utilize the meaningful features by exploiting both spatial and frequency domain. Finally, we use decoder network to obtain finer representation of binary mask to localize tampered region from low-resolution feature maps. In order to develop encoder-decoder network, we utilize convolutional layers, batch normalization, max-pooling and upsampling. Next, we will discuss the technical details of our proposed architecture for image tamper localization.

4.3.1 LSTM Network with Resampling Features

Resampling Features

The typical content-changing manipulations are copy-clone, splicing and object removal, which are difficult to detect. In general, these manipulations distort the natural statistics at the boundary of tampered regions. Fig. 4.3 shows two examples to illustrate the difference in statistics between manipulated and non-manipulated patches. In [21], the method of resampling detection is presented. Laplacian filter along with Radon transform is exploited in order to extract resampling features given a patch. We will also follow the same procedure for resampling features. Given an image, we first extract 64 (8×8) non-overlapping patches. As input image has size of 256x256x3, the dimension of each patch would be 32x32x3. Then, the square root of magnitude of 3×3 Laplacian filter is used to produce the magnitude of linear predictive error for each extracted patch as presented in [21]. As resampling signal has periodic correlations in the linear predictor error, we apply the Radon transform to accumulate errors along various angles of projection. In our experiment, we use 10 angles. Finally, we apply Fast Fourier Transform (FFT) to find the periodic nature of the signal. In general, these resampling features are capable of capturing different resampling characteristics- JPEG quality thresholded above or below a threshold, upsampling, downsampling, rotation clockwise, rotation counterclockwise, and shearing (in an affine transformation matrix).

In order to reduce computational burden, we resize images to 256×256 which might introduce some additional artifacts such as degradation in image quality factor, shearing, upsampling, downsampling. In [21], resampling features are used to classify these artifacts.

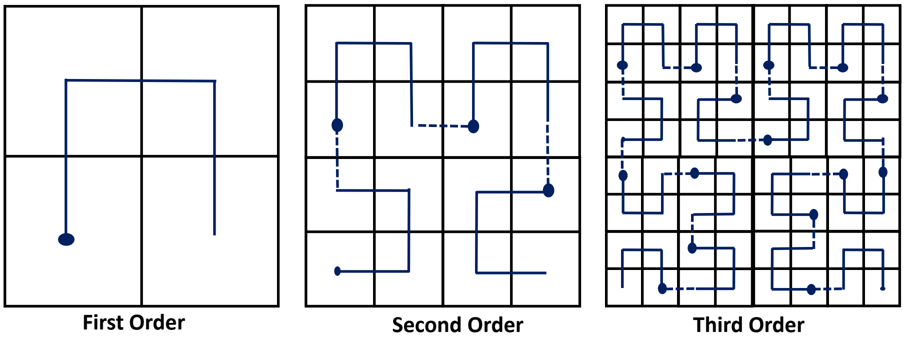


Figure 4.4: The figure illustrates Hilbert curves for different orders.

In this work, we also utilize resampling features, which gives us robust performance. Unlike [21], where resampling features are considered for patch classification, we perform localization at pixel level. There is a tradeoff in selecting the patch size: resampling is more detectable in larger patch sizes because the resampling signal has more repetitions, but small manipulated regions will not be localized that well. In [21], resampling features are extracted from 8 × 8 block. On the other hand, we choose 32 × 32 small patches from an image to extract resampling features that capture more information. The major motivation of utilizing the resampling features for patches is to characterize the local artifacts due to different types of manipulations.

Hilbert Curve

Long-Short Term Memory (LSTM) is commonly used in tasks where sequential information exists. The performance of LSTM highly depends on the ordering of the patches (sequence of the extracted patches). One can consider horizontal or vertical directions, but these orderings do not capture local information well. For example, consider a manipulated object present in the upper-left corner of the image; if we follow horizontal and vertical directions, LSTM will observe different part of the same object after a long interval. Due to this long time lag, LSTM can not correlate well for a region (an object for this case).

In order to preserve better local information, we use space-filling curve which is commonly used to reduce multi-dimensional problem to a one-dimensional [19]. Hilbert curve is one of the space-filling curve methods that traverses the multi-dimensional space linearly, and passes through all points of a square. Fig. 4.4 shows the process of how Hilbert curve works. The main mechanism is to divide a plane into four parts, each of these parts into four parts, and so on. As we have total 64 (8×8) blocks extracted from an image, we require three recursive dividing of the plane. After ordering the patches with Hilbert curve, LSTM network is utilized. We empirically observe that this ordering technique helps improve the performance of localization.

Long-Short Term Memory (LSTM) Network

LSTM network is well-known for processing sequential data in different applications such as language modeling, machine translation, image captioning, and hand writing generation. In computer vision, LSTM network has been successfully used to capture the dependency among a series of pixels [119, 22]. The key insight of using LSTM for detecting image manipulations is *to learn the boundary transformation between different blocks, which provides discriminative features between manipulated and non-manipulated regions.*

In [11, 21], LSTM network is utilized in order to learn the transition (change) between manipulated vs non-manipulated blocks by feeding the blocks into an LSTM network. In [21], the authors propose a patch classification framework where frequency domain features are extracted from 8×8 block before LSTM network. The method could be more effective by considering larger block size. Unlike these approaches, we divide an image into several patches, and extract rasampling features as discussed in Sec. 4.3.1 from 32×32 size of patch that are taken as input to the LSTM network.

After extracting resampling features for each patch, we use Hilbert curve (discussed in Sec. 4.3.1) to determine the ordering of the patches. Then, we feed the resampling features extracted from patches into LSTM cells in a sequential manner. LSTM network computes the logarithmic distance of patch dependency by feeding each patch to each cell. The LSTM cells learn the correlation among neighboring patches. In this framework, we use 2 stacked layers, and at each layer, 64 cells are used. We obtain 64 dimensional feature vector from each cell in the last layer. Then, we project the outputs from LSTM network to N_f features maps. Let us consider feature vector $F_l \in \mathcal{R}^{1 \times N_h}$. If we define a weight matrix W_l ($\in \mathcal{R}^{N_h \times N_f}$), then output feature would be as follows:

$$O_l = F_l.W_l + B_l \tag{4.1}$$

Here, B_l is bias with N_f dimension. For each set of 64 cells, we will obtain $64 \times N_f$ size matrix. The LSTM cells (64 cells) actually provide the transformed feature for each of the patches. Next, we carefully choose the ordering of the cell outputs in order to preserve the spatial information. Then, we reshape the $64 \times N_f$ matrix to $8 \times 8 \times N_f$, where first two dimensions represent the location of the patch.

LSTM Cell Overview. Information flow between the LSTM cells is controlled by three gates: (1) input gate, (2) forget gate, and (3) output gate. Each gate has a value ranging from zero to one, activated by a sigmoid function. Let us denote cell state and output state as C_t and z_t for current cell t. Each cell produces new candidate cell state $\bar{\mathcal{C}_t}$. Using the previous cell state $\mathcal{C}_{t-1}$ and $\bar{C}_t$, we can write the updated cell state $\mathcal{C}_t$ as

$$\mathcal{C}_t = f_t \circ \mathcal{C}_{t-1} + i_t \circ \bar{\mathcal{C}_t} \tag{4.2}$$

Here, $\circ$ denotes the *pointwise* multiplication. Finally, we obtain the output of the current cell h_t, which can be represented as

$$z_t = o_t \circ tanh(\mathcal{C}_t) \tag{4.3}$$

In Eqns. 4.2 and 4.3, i, f, o represent input, forget and output gates.

4.3.2 Encoder Network

Our main objective is to design an efficient architecture for pixel-wise tamper region segmentation. We use convolutional layers to design the encoder which allows the network to understand appearance, shape and the spatial-relationship (context) between manipulated and non-manipulated classes. In [8, 96, 27], some deep architectures are presented where convolutional layers are utilized in order to produce spatial heatmaps for semantic segmentation. As spatial information is very important to localize manipulated regions, we also incorporate convolutional layers into our framework. We exploit and modify encoder-decoder architecture as presented in [8]. The encoder component is similar to CNN architecture except the fully connected layers.

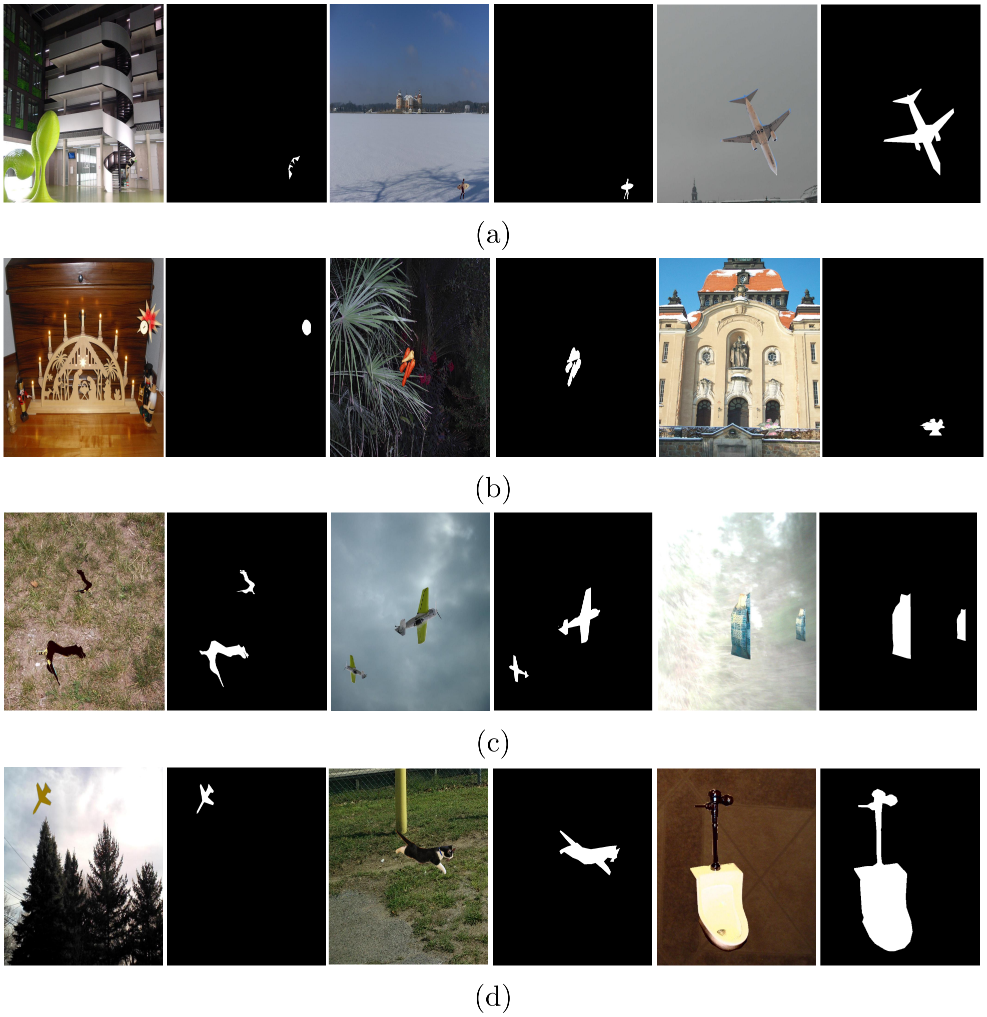
Convolutional Network (ConvNet) consists of different layers, where each layer of data is a three-dimensional array of size $h \times w \times c$, where h and w are height and width of the data, and c is the dimension of the channels. Each layer of convolution involves learnable filters with varying size. The filters in convolutional layer will create feature maps that are connected to the local region of the previous layer. In the first layer, image is taken as input with dimension of $256 \times 256 \times 3$ (width, height, color channels).

The basic building block of each encoder utilizes convolution, pooling, and activation functions. We use residual unit [64] for each encoder. Residual block takes advantage of shortcut connections that are parameter free. The main advantage of using residual unit is that it can easily optimize the residual mapping and more layers are trainable. Let us consider an input to the residual unit is y, and the mapping from input to output of the unit is $\mathcal{T}(.)$. The output of residual unit would be $\mathcal{T}(y) + y$ in the forward pass. In each convolutional layer, we use kernel size of $3 \times 3 \times d$, where d is the depth of a filter. We use different depth for different layers in the network. In encoder network, the number of filters are generally in increasing order. On the other hand, we decrease the number of filters in decoder.

Each residual unit in the encoder produces a set of feature maps. We utilize batch normalization at each convolutional layer. Batch normalization is robust to covariance shift. As an activation function, we choose rectified linear unit (ReLU) that can be represented as $max(0, x)$. At the end of each residual unit, max-pooling with stride 2 is performed, which reduces the size of feature maps by a factor of 2. Unlike [11], we exploit max-pooling at each layer as it provides translation invariance. Each max-pooling operation introduces a loss of spatial resolution (i.e., boundary details) of the feature maps. The loss in boundary detail can be compensated by using decoder which is introduced in [8], and discussed next.

4.3.3 Decoder Network

In [96], a decode technique is proposed that requires encoder feature maps to be stored during prediction. This process might not be applicable in real-life as it requires intensive memory. In this work, we follow a decoding technique that is presented in [8]. In [8], the advantage of using decoder has been discussed in details. The key part is the decoder which replaces the fully connected layers. The decoder decodes the feature output from encoder. As encoder-decoder is primarily developed for semantic object segmentation [8], we exploit and tune this network in order to segment manipulated objects. In the upsampling step, no learnable parameters are involved. Different multi-channel filters are utilized which are convolved with the upsampling heatmaps (coarse representation) to create dense maps. Each decoder follows basic operations - upsample, convolution, and batch normalization. Each decoder first performs upsampling of the feature maps learned at previous layer. Following that, residual unit computation and batch normalization are performed. We use 3×3 size kernel for decoder network. Fig. 4.2 shows the decoder operation of the network. At the end of network, we obtain finer representation of spatial maps that indicates the manipulated regions in an image.



(a)

(b)

(c)

(d)

Figure 4.5: The figures show some manipulated images with corresponding ground-truth masks from synthetic dataset. (a) and (b) shows images created from DRESDEN [54] dataset. (c) and (d) are the manipulated images created from NIST [2] dataset.

4.3.4 Training the Network

Soft-max Layer. In order to predict the pixel-wise classification, softmax layer is used at the end of the network. Let us denote the probability distribution over various classes as $P(\mathcal{Y}_k)$ which is provided by softmax classifier. Now, we can predict label by maximizing $P(\mathcal{Y}_k)$ with respect to k. The predicted label can be obtained by $\hat{\mathcal{Y}} = \arg\max_k \; P(\mathcal{Y}_k)$. As we are only interested to predict manipulated pixels against non-manipulated pixels, the value of k would be 2. Given the predicted mask provided by softmax layer, we can compute the loss that will be used to learn the parameter through back-propagation.

Training Loss. During training, we use cross entropy loss, which is minimized to find the optimal set of parameters of the network. Let θ be the parameter vector corresponding to image tamper localization task. So, the cross entropy loss can be computed as

$$\mathcal{L}(\theta) = -\frac{1}{M}\sum_{m=1}^{M}\sum_{n=1}^{N} \mathbb{1}(\mathcal{Y}^m = n)\log(\mathcal{Y}^m = n | y^m; \theta) \tag{4.4}$$

Here, M and N denote the total number of pixels, and the number of class. y represents the input pixel. $\mathbb{1}(.)$ is an *indicator function*, which equals to 1 if $j = k$, otherwise it equals 0. In our experiment, we observe that weighted cross entropy loss provides better result. It is simply because the imbalance between the number of non-manipulated and manipulated pixels. We put more weight on manipulated pixels over non-manipulated pixels. We use *adaptive moment estimation (Adam)* [75] optimization technique in order to minimize the loss of the network, shown in Eqn. 4.4. At each iteration, one mini-batch is processed to update the parameters of the network. In order to learn the parameters effectively, we choose the mini-batch very carefully which will be discussed in details in Sec. 4.4. After optimizing the loss function over several epochs, we learn the optimal set of parameters of the network. With these optimal parameters, the network is able to predict pixel-wise classification given a test image.

4.4 Experiments

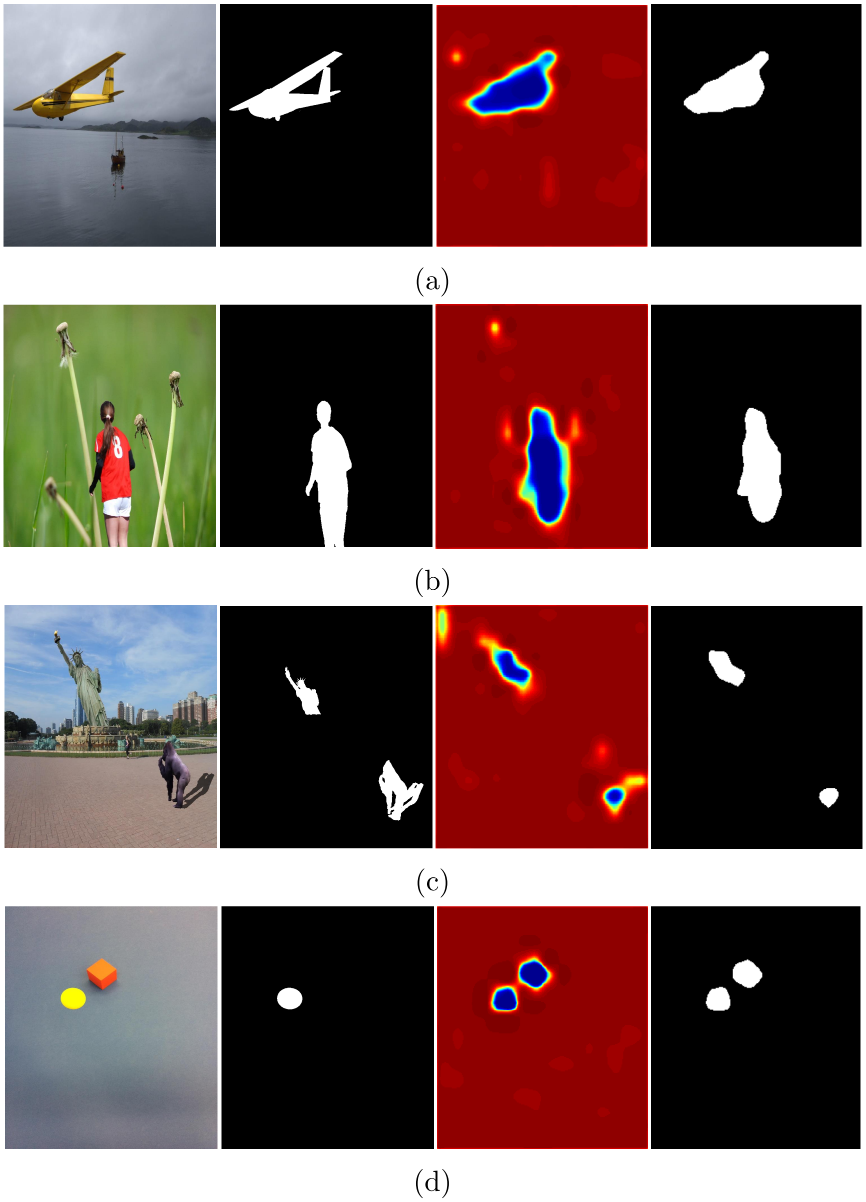
In this section, we demonstrate our experimental results for segmentation of manipulated regions given an image. We evaluate our proposed model on two challenging datasets- NIST'16 [2], and IEEE Forensics Challenge [1].

4.4.1 Datasets

In order to train our proposed architecture, we create a synthesized dataset, which will be discussed in Sec. 4.4.1. In addition, we also use the NIST [2] and IEEE Forensics Challenge [1] datasets to measure the performance.

Creation of Synthesized Data

As deep learning networks are extremely data hungry, there is a need to collect images for training and testing the networks. For training, we will need plentiful examples (usually tens of thousands) of both manipulated and non-manipulated images. Towards this goal, we create approximately $65k$ manipulated images in order to train the proposed network discussed in Sec. 4.3. This network will be referred to as 'Base-Model'. The 'Base-Model' will then be fine-tuned with the NIST [2] and IEEE Forensics Challenge [1] datasets. Below



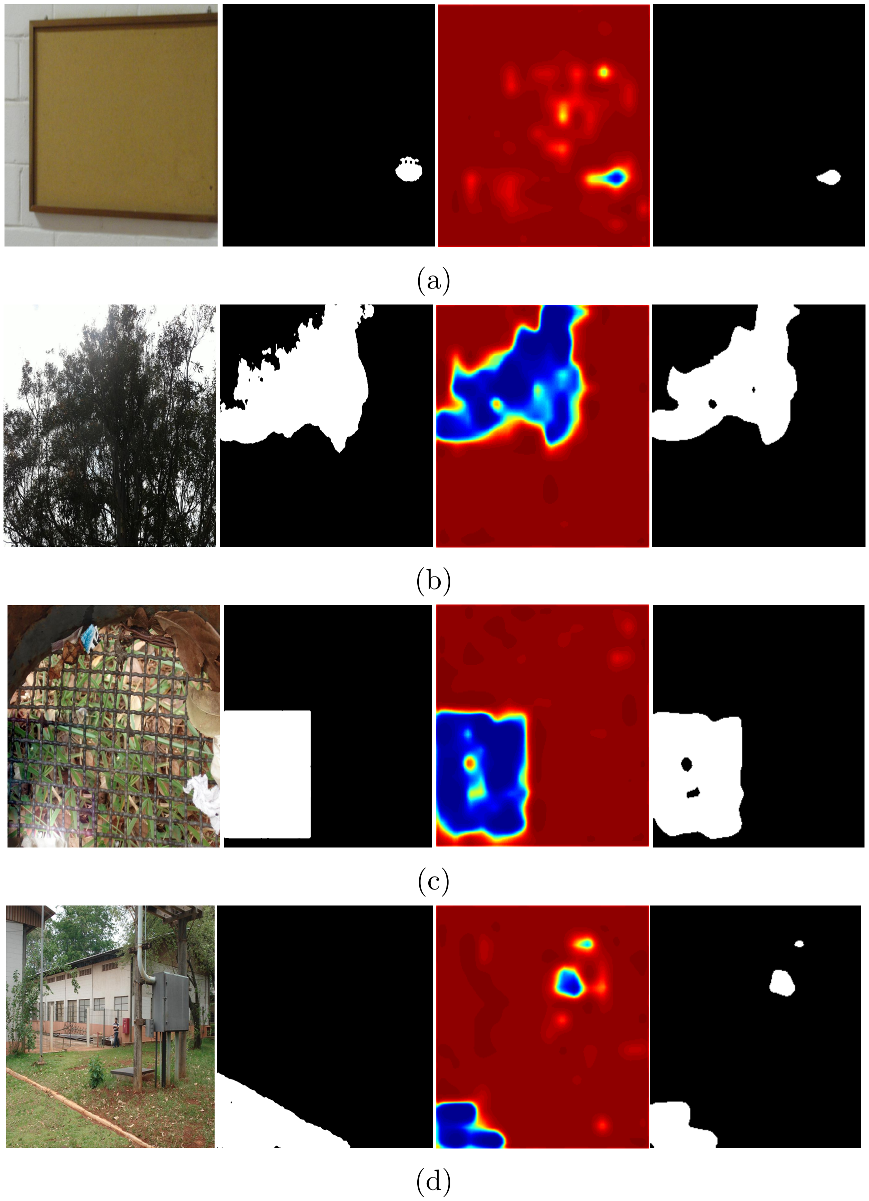
(a)

(b)

(c)

(d)

Figure 4.6: This figure illustrates some segmentation results on NIST [2] dataset. First and second columns represent input image and ground-truth mask for tampered region. Third and fourth columns delineate probability heat map and predicted binary mask.



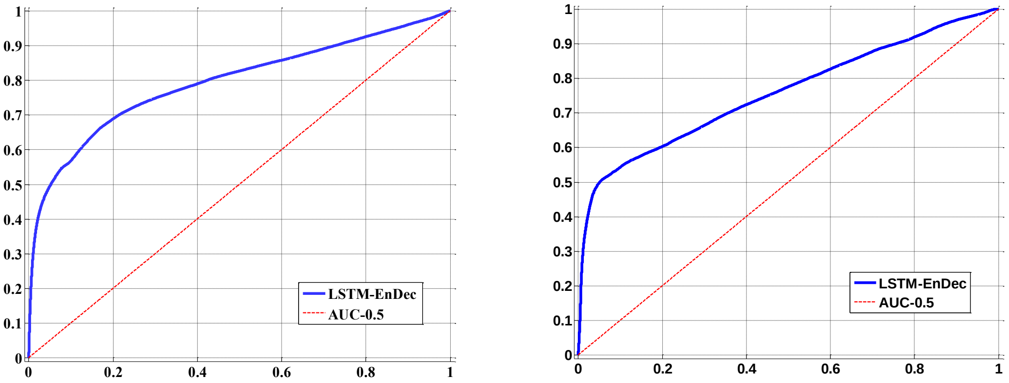
(a)

(b)

(c)

(d)

Figure 4.7: Some segmentation examples on IEEE Forensics Challenge [1] dataset are shown in this figure. First and second columns are input images and ground-truth masks for manipulated regions. Third and fourth columns demonstrate the probability heatmap and predicted binary mask.



(a) NIST'16 [2]

(b) IEEE Forensics Challenge [1]

Figure 4.8: The figures demonstrate ROC plots on NIST'16 [2], and IEEE Forensics Challenge [1] datasets respectively. Each curve has area under the curve (AUC), which are provided in Table 4.4

we explain the innovation in the collection of the manipulated image set.

In the synthesized dataset, we have focused on mainly object splicing (additions/subtractions) manipulation. The major challenge of creating manipulated images was to obtain segmented objects to insert into an image. For this we used the MS-COCO dataset [90], which is largely used for object detection and semantic segmentation, to obtain segmented objects across a variety of categories. We extracted the objects from MS-COCO[90] images using image masks provided in ground-truth. Finally, these objects are used to create manipulation from the images of DRESDEN dataset[54] and NIST dataset[2]. Please note that we use only non-manipulated images from NIST dataset to create manipulation.

To create a new manipulated image, we followed the steps below.

(1) For each raw image in the DRESDEN dataset[54], we cropped each of the image's corners to extract a 1024x1024 patch. This method avoids resizing which introduces additional image distortions.

Data Set	# image pairs	Avg. Image Size
CoMoFod[141]	260	512x512
Manip[31]	48	2305x3020
GRIP[36]	100	1024x786
COVERAGE[150]	100	400x486
Synthesized	65k	1024x1024

Table 4.1: A comparison of common image tampering datasets

(2) For each of these image patches we spliced on six different objects, from the MS-COCO dataset, to create six splice manipulated images.

(3) In order to create diverse splicing data, we spliced the same object onto the patch twice with different scaling and rotation factor, while ensuring no overlap as shown in Fig. 4.5. This entire process was automated allowing us to generate tens of thousands of images in less than a day with no human interaction. Using the Dresden image database as the source of non-manipulated images we were able to produce approximately $40k$ images and an additional $25k$ using the DRESDEN and NIST datasets respectively. The scale of our data is a hundred fold increase over most datasets that offer similar types of manipulations, which allows us to train a deep learning model. Our synthesized data also has a relatively high resolution. We can see how our dataset compares to similar datasets in table 4.1. With this newly generated data, we trained the 'Base-Model'. The base model predicts manipulated region at pixel level given an image.

Dataset Preparation

In order to evaluate our model, we chose two datasets which provided ground-truth mask for manipulated regions. NIST'16 [2] is a very challenging dataset, which includes

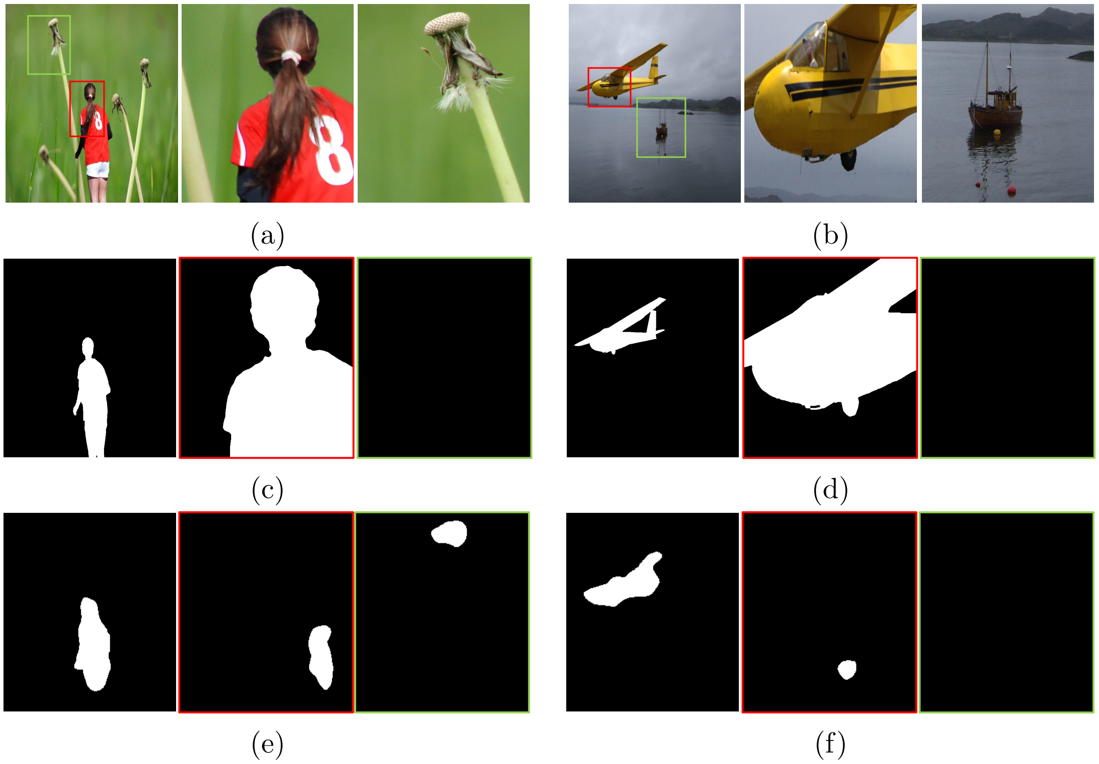


Figure 4.9: This figure demonstrates the segmentation performance with patches as input on NIST'16 [2] dataset. First column of (a) represents the input image. Second and third columns of (a) delineate the patches as shown in the bounding boxes of input image (first column). Figures (c,d) and (e,f) are corresponding ground-truth mask and predicted binary mask.

three main types of manipulation - (a) copy-clone, (b) removal, and (c) splicing. This recently released dataset includes images, which are tampered in a sophisticated way to beat current state-of-the-art detection techniques. We also show our results on the IEEE Forensics Challenge [1] dataset which provides ground-truth mask for manipulation. As manipulated regions are small in number compared to non-manipulated regions, we also perform data augmentation in order to get rid of bias in training.

In data preparation, we first split the whole image dataset into three subsets-training (70%), validation (5%) and testing (25%). These subsets are chosen randomly. In order to increase the training data, we extract bigger patches from the four corners of the image. One additional patch is also extracted from center location of the image. We crop patches with size 1024 × 1024 from NIST'16 [2] training images to optimize the parameters of our architecture. The spatial resolution of IEEE Forensics Challenge [1] dataset is comparatively low. So, we extract 512 × 512 size of patches for IEEE Forensics Challenge [1] dataset. These newly generated images usually contain partial manipulated objects when compared to original images. We only perform data augmentation on training set, not in validation and test set. As the image and corresponding ground-truth mask are the same size, we can easily generate the ground-truth masks for the extracted image patches. With these newly generated ground-truth masks and patches, we train the whole network end-to-end.

4.4.2 Experimental Analysis

In this section, we will discuss the implementation and evaluation criterion of our model. We compare our model with different existing methods for image forgery detection.

Implementation Details. We implement our proposed framework in TensorFlow. In order to expedite our computational load, we utilize multi-GPU setting. We use two NVIDIA Tesla $K80$ GPUs to perform different sets of experiments, which will be discussed next.

Evaluation Criterion. In order to evaluate our model, we use pixel-wise accuracy and receiver operating characteristic (ROC) curve. ROC curve measures the performance of binary classification task by varying the threshold on prediction score. The area under the ROC curve (AUC) is computed from the ROC curve that measures the distinguishable ability of a system for binary classification. The AUC value typically lies in between 0 and 1.0. The AUC with 1.0 is sometimes referred as perfect system (no false alarm).

Experimental Setup. We setup few experiments to evaluate our proposed architecture. They are (1) performance of the proposed model, (2) performance with different baseline methods, (3) comparison against existing state-of-the-art approaches, (4) ROC curve, (5) qualitative analysis, and (6) impact of global context.

Baseline Methods: In this section, we will introduce some baseline methods. We implement and compare against these methods. The various baseline methods are described below.

⋄ *FCN* : Fully convolutional network as presented in [96].

⋄ *J-Conv-LSTM-Conv*: This method utilizes LSTM network and convolutional layers for segmentation as in [11].

⋄ *Encoder-Decoder*: This method utilizes convolutional network as encoder and deconvolution as decoder, proposed in [8].

Methods	NIST [2]	IEEE [1]
FCN [96]	74.28%	–
Encoder-DeCoder [8]	82.96%	-
J-Conv-LSTM-Conv [11]	84.60%	77.67%
LSTM-EnDec-Base	91.36%	88.24%
LSTM-EnDec	**94.80%**	**91.19%**

Table 4.2: The table shows the pixel-wise accuracy on NIST [2], IEEE Forensics Challenge [1] datasets for image tamper segmentation.

◇ *EnDec*: Similar to encoder-decoder [8] with upsampling factor of 4 in deconvolution.

◇ *LSTM-EnDec-Base*: Proposed architecture as shown in Fig. 4.2 trained on Synthesized dataset discussed in Sec. 4.5

◇ *LSTM-EnDec*:Finetuned model of proposed architecture as shown in Fig. 4.2

Performance of the Proposed Model.

We test our proposed model on two datasets- NIST'16 [2], IEEE Forensics Challenge [1]. We first train our model with synthesized data (discussed in Sec. 4.5). We refer this model as 'LSTM-EnDec-Base' model. The LSTM-EnDec-Base model is finetuned with training sets from NIST'16 [2], IEEE Forensics Challenge [1] datasets. We obtain two finetuned model for two datasets. Table 4.2 shows pixel-wise classification accuracy on segmentation task. 'LSTM-EnDec-Base' model learns good discriminative properties between manipulated vs non-manipulated pixels. Finally, finetuning this 'LSTM-EnDec-Base' model provides a boost in performance for labeling tamper class at pixel level. From the table, we can see that proposed model 'LSTM-EnDec' outperforms 'LSTM-EnDec-Base' model by 3.44%, and 2.95% on NIST'16 [2], IEEE Forensics Challenge [1] datasets respectively.

Performance with Different Baseline Methods.

In semantic segmentation, some recent architectures such as fully convolutional netowork (FCN) [96] and Encoder-Decoder (SegNet) [8] have successfully exploited. For comparison, we implement and train these deep architectures with image manipulation data to compare the performance of our model. We can see from Table. 4.2 that convolutional neural network based model such as FCN, and SegNet does not perform well compared to proposed architecture for tamper localization. It is because these models try to learn the visual concept/feature from an image whereas manipulation of an image does not leave any visual clue. We empirically observe that FCN and SegNet prone to misclassify for copy-clone and object removal type of manipulations. LSTM-EnDec surpasses FCN and Encoder-Decoder network by 20.52% and 11.84% on NIST [2] as shown in Table. 4.2. We also compare against the segmentation framework for tamper localization (J-Conv-LSTM-Conv) presented in [11]. The proposed network outperforms J-Conv-LSTM-Conv by large margin. The advantage of our proposed model over J-Conv-LSTM-Conv is that proposed model can learn larger context by exploiting correlation between patches. On the other hand, J-Conv-LSTM-Conv is limited to correlate between different blocks of a patch. The exploitation of both LSTM network with resampling features and spatial features using encoder, helps the overall architecture to learn manipulations better.

Comparison against Existing Approaches.

In media forensics, there have been lot of approaches presented for image tamper localization. Some of them are DCT Histograms [91], ADJPEG [17], NADJPEG [17],

Methods	AUC score
DCT Histograms [91]	0.545
ADJPEG [17]	0.5891
NADJPEG [17]	0.6567
PatchMatch [36]	0.6513
Error level analysis [99]	0.4288
Block Features [82]	0.4785
Noise Inconsistencies [103]	0.4874
LSTM-EnDec	**0.7936**

Table 4.3: Comparison against existing approaches on NIST [2] dataset.

PatchMatch [36], Error level analysis [99], Block Features [82], and Noise Inconsistencies [103]. Table. 4.3 shows the comparison against some existing state-of-the-art methods for image tamper localization. From the table, we can observe that our framework outperforms other existing methods by large margin on NIST'16 [2] dataset.

ROC Curve.

Figs. 4.8(a,b) show the ROC plots for image tamper localization, on NIST'16 [2], and IEEE Forensics Challenge [1], datasets respectively. These ROC curves measure the performance of binary pixel classification whether a pixel is manipulated or not. We also provide the area under the curve (AUC) results in Table 4.4. Our model achieves AUC of 0.7936 and 0.7577 on NIST and IEEE Forensics datasets respectively. From the ROC curves as shown in Figs. 4.8(a) and 4.8(b), we can see that the proposed network classifies tampered pixels with high confidence.

Dataset	Segmentation
NIST [2]	0.7936
IEEE Forensic [1]	0.7577

Table 4.4: Area under the curve (AUC) for the ROC plots as shown in Fig. 4.8

Qualitative Analysis of Segmentation.

In Figs. 4.6 and 4.7, we provide some examples showing segmentation results produced by the proposed network. Fig. 4.6 shows segmentation results on NIST'16 [2] dataset. Segmentation results for IEEE Forensics Challenge [1] dataset are illustrated in Fig. 4.7. We also provide probability heat map for localizing tampered region as shown in third column of Figs. 4.6 and 4.7. As we can see from the Figs. 4.6 and 4.7, the predicted mask can locate manipulated regions from an image with high probability. The boundary of tampered objects is affected in the segmentation results as shown in Fig. 4.6 (third column), the underlying reason being that image boundaries are smooth (blurred) for NIST'16 [2] dataset. However, our proposed network can still localize precisely with higher overlap compared to ground-truth mask.

Impact of Global Context.

In our framework, we consider images as input so that the network can exploit global context. In order to observe the effectiveness of global context, we run an experiment where we consider patches as input to the network instead of images. Fig. 4.9 illustrates the segmentation results with respond to the input patches on NIST [2] dataset. From the figure, we can see that the network can localize more precisely given an image. On the other hand, the precision of localization degrades for smaller patch as the patch misses the broader context. In case of manipulated patch as shown in Figs. 4.9(a) and 4.9(b) (middle column), proposed network detects the part of the manipulated objects. For example, *digit of the person's dress* and *wheel of a plane* are identified as manipulated as shown in Figs. 4.9(e) and 4.9(f) respectively. For the patch with non-manipulated pixels, the network may provide false alarm sometimes as demonstrated in Figs. 4.9(e) (third column). From this study, we can conclude that global context helps analyzing the manipulated images.

Dataset	Segmentation
LSTM-EnDec-Image	94.80%
LSTM-EnDec-Patch	89.98%

Table 4.5: The table illustrates the pixel-wise accuracy for the proposed architecture with image and patch as input on NIST [2] dataset.

Table 4.5 shows the accuracy of pixel labeling of the proposed model for identifying the image manipulation with different set of inputs (image and patch). We extract 16 patches from each image of the test set in order to evaluate the performance by utilizing the sliding window approach. From the table, we can observe that the performance of LSTM-EnDec degrades when patches are taken as input since the network could not exploit global context. On the other hand, LSTM-EnDec demonstrates higher performance with images as input to the network. LSTM-EnDec-Image outperforms LSTM-EnDec-Patch by large margin (**4.82**%) in recognizing manipulated pixels as shown in Table 4.5.

4.5 Conclusion

In this chapter, we present a deep learning based approach to semantically segment manipulated regions in a tampered image. In particular, we employ a hybrid CNN-LSTM model that effectively classifies manipulated and non-manipulated regions. We exploit CNN architecture to design an encoder network that provides spatial feature maps of manipulated objects. Resampling features of the patches are incorporated in LSTM network to observe the transition between manipulated and non-manipulated patches. Finally, a decoder network is used to learn the mapping from encoded feature maps to binary mask. Furthermore, we also present a new synthesized dataset which includes large number of images. Our detailed experiments showed that our approach could efficiently segment various types of manipulations including copy-move, object removal and splicing.

Chapter 5

Conclusions

5.1 Thesis Summary

In this thesis, we addressed two fundamental challenges currently facing image and video analysis approaches - how to minimize the labeling effort by choosing informative samples to label, and how to identify image regions that have been manipulated. In the first two frameworks, we proposed novel information-theoretic strategies to select the most informative samples for manual labeling. The main purpose of these approaches is to learn a good recognition model with reduced human annotation effort. In our final approach, we proposed a deep neural network in order to localize manipulated objects from an image. The proposed network can automatically segment out manipulated regions from non-manipulated one.

In first approach as discussed in Chapter 2, we proposed a novel active learning framework for joint scene and object classification where inter-dependencies between scene and object samples had been exploited. We demonstrated how joint entropy of a graph formulated from scene and object samples could be reduced by utilizing the mutual information between nodes of the graph. With only a small subset of the full training set, our approach achieved better or similar performance compared with the model trained on full training set. In Chapter 3, we introduced the notion of typicality which could be used as an important tool to learn informative samples from a huge pool of unlabeled samples. Typicality efficiently links between recognition and temporal relationship model. This method is computationally faster than first approach for sample selection task. We applied our method on different computer vision problems such as sample selection, and anomaly detection, which has a great use in different video analysis tasks. This method reduced the human load in labeling samples for visual recognition tasks without compromising the performance compared to the model with 100% labeling. Our method also efficiently detects the anomalous sample from a video.

In Chapter 4, we presented our final method which exploited deep neural network for localizing manipulated regions from an image. In this network, we designed a novel architecture utilizing resampling features, CNN and LSTM networks in order to effectively classify manipulated pixels given an image. CNN architecture provides spatial feature maps of manipulated objects. The LSTM network observes the transition between manipulated and non-manipulated patches by extracting resampling features from the patches of an image. Finally, we exploited a decoder network to learn the mapping from encoded feature maps to fine grained binary mask. In this chapter, in order to train deep network, we also proposed a new synthesized dataset which includes large number of images. This dataset could be beneficial to media forensics community, especially if one wants to train a deep network. We showed rigorous experiments where our approach could efficiently segment various types of manipulations.

5.2 Future Research Directions

Our research aims to explore advanced machine learning techniques and information-theoretic approaches to solve various computer vision problems. We presented two novel active learning strategies that were applied to scene classification, object detection, and action recognition. In this thesis, we restricted our methods to label the class of a sample for any recognition task. However, classifications tasks such as object detection, and activity recognition require the location ground-truth in an image along with class label for a sample. At training phase, a model is required to access the bounding box information and the class labels in order to efficiently detect an object. Similarly, it is true for activity recognition which requires the information of spatio-temporal location of an activity. The sample selection strategy for the tasks like object detection and activity recognition where the location and the class label of sample are required to be manually labeled, will be a challenging future research direction.

In computer vision, transfer learning has become a growing interest where the models learned in one task can be used in another task with little additional supervision. In Chapter 3, we exploited typicality to measure the informativeness of the samples. As a future direction, typicality can be utilized to transfer knowledge from one domain where data is available to another where there is limited labeled data. One can learn a good representation of recognition model when there are enough data samples. However, transferring some information from this learned representation to an unknown domain will be an interesting research problem. Typicality can be used as powerful tool to measure the relevancy of two domains where one is known and the other will be learned. Furthermore, the possible future application of our active learning frameworks presented in this thesis might be to adjust the parameters of a deep network with a set of unlabeled data.

In Chapter 4, we presented deep learning based approaches in order to localize manipulated regions from an image. The localization of manipulated objects has diverse applications, especially in security and surveillance. In our approaches, we considered pixel correlations, frequency domain features and low-level features (e.g., edges, and textures) to localize tamper regions. The high level information such as contextual regularity for detecting manipulated objects can be a potential future direction. For example, a car is more inclined to appear in road scene than indoor scenes. If a car is spliced into 'indoor' image, the context model should detect this anomaly. Another possible future direction is to exploit deep neural network to identify manipulation in videos.

Another future research direction of this thesis could be the active learning for adversarial examples. The adversarial examples are created by perturbing the original samples (e.g., images or videos) in a controlled way which causes the recognition model to make a mistake. In active learning methods, we assume that the human or Oracle will always provide the correct labeling. However, in the context of adversarial examples, human/Oracle may provide inaccurate labels that may corrupt the learning process of a recognition model. Designing a robust active learning framework for adversarial examples will be an interesting research problem in the future.

Bibliography

[1] IEEE IFS-TC Image Forensics Challenge Dataset. `http://ifc.recod.ic.unicamp.br/fc.website/index.py`.

[2] NIST Nimble 2016 Datasets. `https://www.nist.gov/sites/default/files/documents/2016/11/30/should_i_believe_or_not.pdf`.

[3] Stationary distributions of markov chains. `https://brilliant.org/wiki/stationary-distributions/`. Retrieved: May 10, 2018.

[4] Osamah M Al-Qershi and Bee Ee Khoo. Passive detection of copy-move forgery in digital images: State-of-the-art. *Forensic science international*, 231(1):284–295, 2013.

[5] Marina Alberti, John Folkesson, and Patric Jensfelt. Relational approaches for joint object classification and scene similarity measurement in indoor environments. In *AAAI 2014 Spring Symposia: Qualitative Representations for Robots*, 2014.

[6] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, 6(3):1099–1110, 2011.

[7] Mohd Dilshad Ansari, Satya Prakash Ghrera, and Vipin Tyagi. Pixel-based image forgery detection: A review. *IETE journal of education*, 55(1):40–46, 2014.

[8] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[9] Jawadul H. Bappy, Sujoy Paul, and Amit Roy-Chowdhury. Online adaptation for joint scene and object classification. In *ECCV*, 2016.

[10] Jawadul H Bappy, Sujoy Paul, Ertem Tuncel, and Amit K Roy-Chowdhury. The impact of typicality for informative representative selection. *Computer Vision and Pattern Recognition (CVPR).*, 2017.

[11] Jawadul H Bappy, Amit K Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and BS Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *ICCV*, 2017.

[12] Belhassen Bayar and Matthew C Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, pages 5–10, 2016.

[13] Belhassen Bayar and Matthew C Stamm. Design principles of convolutional neural networks for multimedia forensics. In *IS&T International Symposium on Electronic Imaging: Media Watermarking, Security, and Forensics*, 2017.

[14] Belhassen Bayar and Matthew C Stamm. On the robustness of constrained convolutional neural networks to jpeg post-compression for image resampling detection. In *Proceedings of The 42nd IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017.

[15] Aparna Bharati, Richa Singh, Mayank Vatsa, and Kevin W Bowyer. Detecting facial retouching using supervised deep learning. *IEEE Transactions on Information Forensics and Security*, 11(9):1903–1913, 2016.

[16] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved dct coefficient analysis for forgery localization in jpeg images. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2011.

[17] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012.

[18] Luca Bondi, Silvia Lameri, David Güera, Paolo Bestagini, Edward J Delp, and Stefano Tubaro. Tampering detection and localization through clustering of camera-based cnn features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1855–1864, 2017.

[19] Greg Breinholt and Christoph Schierz. Algorithm 781: generating hilbert's space-filling curve by recursion. *ACM Transactions on Mathematical Software (TOMS)*, 24(2):184–189, 1998.

[20] Michele Buccoli, Paolo Bestagini, Massimiliano Zanoni, Augusto Sarti, and Stefano Tubaro. Unsupervised feature learning for bootleg detection using deep learning architectures. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014.

[21] Jason Bunk, Jawadul H Bappy, Tajuddin Manhar Mohammed, Lakshmanan Nataraj, Arjuna Flenner, BS Manjunath, Shivkumar Chandrasekaran, Amit K Roy-Chowdhury, and Lawrence Peterson. Detection and localization of image forgeries using resampling features and deep learning. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1881–1889, 2017.

[22] Wonmin Byeon, Thomas M Breuel, Federico Raue, and Marcus Liwicki. Scene labeling with lstm recurrent neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[23] Yanjun Cao, Tiegang Gao, Li Fan, and Qunting Yang. A robust detection algorithm for copy-move forgery in digital images. *Forensic science international*, 214(1):33–43, 2012.

[24] Chih-Chung Chang and Chih-Jen Lin. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

[25] I-Cheng Chang, J Cloud Yu, and Chih-Chuan Chang. A forgery detection algorithm for exemplar-based inpainting images using multi-region relation. *Image and Vision Computing*, 31(1):57–71, 2013.

[26] Jiansheng Chen, Xiangui Kang, Ye Liu, and Z Jane Wang. Median filtering forensics based on convolutional neural networks. *IEEE Signal Processing Letters*, 22(11):1849–1853, 2015.

[27] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[28] Kai-Wen Cheng, Yie-Tarng Chen, and Wen-Hsien Fang. Video anomaly detection and localization using hierarchical feature representation and gaussian process regression. In *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*, pages 2909–2917. IEEE, 2015.

[29] Myung Jin Choi, Joseph J Lim, Antonio Torralba, and Alan S Willsky. Exploiting hierarchical context on a large database of object categories. In *CVPR*, 2010.

[30] Wongun Choi, Khuram Shahid, and Silvio Savarese. Learning context for collective activity recognition. In *CVPR*, pages 3273–3280, 2011.

[31] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou. An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on Information Forensics and Security*, 7(6):1841–1854, Dec 2012.

[32] Yang Cong, Junsong Yuan, and Ji Liu. Sparse reconstruction cost for abnormal event detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3449–3456. IEEE, 2011.

[33] Thomas M Cover and Joy A Thomas. Elements of information theory 2nd edition. 2006.

[34] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[35] D. Cozzolino, G. Poggi, and L. Verdoliva. Efficient dense-field copy;move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11):2284–2297, Nov 2015.

[36] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Efficient dense-field copy–move forgery detection. *IEEE Transactions on Information Forensics and Security*, 10(11):2284–2297, 2015.

[37] IBM ILOG CPLEX. V12. 1: User's manual for cplex. *International Business Machines Corporation*, 46(53):157, 2009.

[38] Pieter-Tjerk De Boer, Dirk P Kroese, Shie Mannor, and Reuven Y Rubinstein. A tutorial on the cross-entropy method. *Annals of operations research*, 2005.

[39] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[40] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Mid-level visual element discovery as discriminative mode seeking. In *NIPS*, 2013.

[41] Gregory Druck, Burr Settles, and Andrew McCallum. Active learning by labeling features. In *EMNLP*, 2009.

[42] Krista A Ehinger, Antonio Torralba, and Aude Oliva. A taxonomy of visual scenes: Typicality ratings and hierarchical classification. *Journal of Vision*, 10(7):1237–1237, 2010.

[43] Ehsan Elhamifar, Guillermo Sapiro, Allen Yang, and S Sasrty. A convex optimization framework for active learning. In *ICCV*, 2013.

[44] Wei Fan, Kai Wang, and François Cayre. General-purpose image forensics using patch likelihood under image statistical models. In *Information Forensics and Security (WIFS), 2015 IEEE International Workshop on*, pages 1–6, 2015.

[45] Hany Farid. Exposing digital forgeries from jpeg ghosts. *IEEE transactions on information forensics and security*, 4(1):154–160, 2009.

[46] Alireza Fathi, Maria Florina Balcan, Xiaofeng Ren, and James M Rehg. Combining self training and active learning for video segmentation. In *BMVC*, volume 29, pages 78–1, 2011.

[47] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 32(9):1627–1645, 2010.

[48] Xiaoying Feng, Ingemar J Cox, and Gwenaël Doërr. An energy-based method for the forensic detection of re-sampled images. In *IEEE International Conference on Multimedia and Expo (ICME)*, 2011.

[49] Xiaoying Feng, Ingemar J Cox, and Gwenael Doerr. Normalized energy density-based forensic detection of resampled images. *IEEE Transactions on Multimedia*, 14(3):536–545, 2012.

[50] Pasquale Ferrara, Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Image forgery localization via fine-grained analysis of cfa artifacts. *IEEE Transactions on Information Forensics and Security*, 7(5):1566–1577, 2012.

[51] Claude Fillion and Gaurav Sharma. Detecting content adaptive scaling of images for forensic applications. In *Media Forensics and Security*, 2010.

[52] Ross Girshick. Fast r-cnn. In *ICCV*, 2015.

[53] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jagannath Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014.

[54] Thomas Gloe and Rainer Böhme. The dresden image database for benchmarking digital image forensics. *Journal of Digital Forensic Practice*, 3(2-4):150–159, 2010.

[55] S Alireza Golestaneh and Damon M Chandler. Algorithm for jpeg artifact reduction via local edge regeneration. *Journal of Electronic Imaging*, 23(1):013018–013018, 2014.

[56] Yunchao Gong, Liwei Wang, Ruiqi Guo, and Svetlana Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *ECCV*. 2014.

[57] Christine Guillemot and Olivier Le Meur. Image inpainting: Overview and recent advances. *Signal processing magazine*, 31(1):127–144, 2014.

[58] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 733–742. IEEE, 2016.

[59] Mahmudul Hasan and Amit Roy-Chowdhury. Incremental activity modeling and recognition in streaming videos. In *CVPR*, 2014.

[60] Mahmudul Hasan and Amit K Roy-Chowdhury. Context aware active learning of activity recognition models. In *ICCV*, 2015.

[61] Mohammad Farukh Hashmi, Vijay Anand, and Avinas G Keskar. Copy-move image forgery detection using an efficient and robust method combining un-decimated wavelet transform and scale invariant feature transform. *AASRI Procedia*, 9:84–91, 2014.

[62] Munawar Hayat, Salman H Khan, Mohammed Bennamoun, and Senjian An. A spatial layout and scale invariant feature representation for indoor scene classification. *arXiv preprint arXiv:1506.05532*, 2015.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *ECCV 2014*, pages 346–361. 2014.

[64] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[65] Anders Høst-Madsen, Elyas Sabeti, and Chad Walton. Data discovery and anomaly detection using atypicality: Theory. *arXiv preprint arXiv:1709.03189*, 2017.

[66] Xia Hu, Jiliang Tang, Huiji Gao, and Huan Liu. Actnet: Active learning for networked texts in microblogging. In *SDM*, pages 306–314. SIAM, 2013.

[67] Maryam Jaberi, George Bebis, Muhammad Hussain, and Ghulam Muhammad. Accurate and robust localization of duplicated region in copy–move image forgery. *Machine vision and applications*, 25(2):451–475, 2014.

[68] Rani Mariya Joseph and AS Chithra. Literature survey on image manipulation detection. *International Research Journal of Engineering and Technology (IRJET)*, 2(04), 2015.

[69] Christoph Kading, Alexander Freytag, Erik Rodner, Paul Bodesheim, and Joachim Denzler. Active learning and discovery of object categories in the presence of unnameable instances. In *CVPR*, 2015.

[70] Christoph Käding, Alexander Freytag, Erik Rodner, Andrea Perino, and Joachim Denzler. Large-scale active learning with approximations of expected model output changes. In *German Conference on Pattern Recognition*, pages 179–191, 2016.

[71] Pravin Kakar and N Sudha. Exposing postprocessed copy–paste forgeries through transform-invariant features. *IEEE Transactions on Information Forensics and Security*, 7(3):1018–1028, 2012.

[72] Ashish Kapoor, Kristen Grauman, Raquel Urtasun, and Trevor Darrell. Active learning with gaussian processes for object categorization. In *ICCV*, 2007.

[73] Alex Kendall, Vijay Badrinarayanan, and Roberto Cipolla. Bayesian segnet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding. *arXiv preprint arXiv:1511.02680*, 2015.

[74] Jaechul Kim and Kristen Grauman. Observe locally, infer globally: a space-time mrf for detecting abnormal activities with incremental updates. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2921–2928. IEEE, 2009.

[75] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[76] Younghee Kwon, Kwang In Kim, James Tompkin, Jin Hyung Kim, and Christian Theobalt. Efficient learning of image super-resolution and compression artifact removal with semi-local gaussian processes. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1792–1805, 2015.

[77] Agata Lapedriza, Hamed Pirsiavash, Zoya Bylinskii, and Antonio Torralba. Are all training examples equally valuable? *arXiv preprint arXiv:1311.6510*, 2013.

[78] Guohui Li, Qiong Wu, Dan Tu, and Shaojie Sun. A sorted neighborhood approach for detecting duplicated regions in image forgeries based on dwt and svd. In *IEEE International Conference on Multimedia and Expo*, 2007.

[79] Haodong Li, Weiqi Luo, Xiaoqing Qiu, and Jiwu Huang. Image forgery localization via integrating tampering possibility maps. *IEEE Transactions on Information Forensics and Security*, 12(5):1240–1252, 2017.

[80] Jian Li, Xiaolong Li, Bin Yang, and Xingming Sun. Segmentation-based image copy-move forgery detection scheme. *IEEE Transactions on Information Forensics and Security*, 10(3):507–518, 2015.

[81] Jun Li, José M Bioucas-Dias, and Antonio Plaza. Spectral–spatial classification of hyperspectral data using loopy belief propagation and active learning. *Geoscience and Remote Sensing, IEEE Transactions on*, 51(2):844–856, 2013.

[82] Weihai Li, Yuan Yuan, and Nenghai Yu. Passive detection of doctored jpeg image via block artifact grid extraction. *Signal Processing*, 89(9):1821–1829, 2009.

[83] Xianglin Li, Runqiu Guo, and Jun Cheng. Incorporating incremental and active learning for scene classification. In *ICMLA*, 2012.

[84] Xin Li and Yuhong Guo. Adaptive active learning for image classification. In *CVPR*, 2013.

[85] Xin Li and Yuhong Guo. Multi-level adaptive active learning for scene classification. In *ECCV*. 2014.

[86] Yuan Li and Ram Nevatia. Key object driven multi-category object recognition, localization and tracking using spatio-temporal context. In *ECCV*, 2008.

[87] Zhicheng Li and Laurent Itti. Saliency and gist features for target detection in satellite images. *TIP*, 20(7):2017–2029, 2011.

[88] Zaoshan Liang, Gaobo Yang, Xiangling Ding, and Leida Li. An efficient forgery detection algorithm for object removal by exemplar-based image inpainting. *Journal of Visual Communication and Image Representation*, 30:75–85, 2015.

[89] Sung Hoon Lim, Chien-Yi Wang, and Michael Gastpar. Information-theoretic caching: The multi-user case. *IEEE Transactions on Information Theory*, 63(11):7018–7037, 2017.

[90] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[91] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492–2501, 2009.

[92] C Liu, J Yuen, and A Torralba. Dense scene alignment using sift flow for object recognition. In *CVPR*, 2009.

[93] Ming-Yu Liu, Oncel Tuzel, Srikumar Ramalingam, and Rama Chellappa. Entropy rate superpixel segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2097–2104. IEEE, 2011.

[94] Qingzhong Liu and Zhongxue Chen. Improved approaches with calibrated neighboring joint density to steganalysis and seam-carved forgery detection in jpeg images. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(4):63, 2015.

[95] Yaqi Liu, Qingxiao Guan, Xianfeng Zhao, and Yun Cao. Image forgery localization based on multi-scale convolutional neural networks. *arXiv preprint arXiv:1706.07842*, 2017.

[96] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[97] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2720–2727. IEEE, 2013.

[98] Weiqi Luo, Jiwu Huang, and Guoping Qiu. Robust detection of region-duplication forgery in digital image. In *18th International Conference on Pattern Recognition*, 2006.

[99] Weiqi Luo, Jiwu Huang, and Guoping Qiu. Jpeg error analysis and its applications to digital image forensics. *IEEE Transactions on Information Forensics and Security*, 5(3):480–491, 2010.

[100] Oisin Mac Aodha, Neill Campbell, Jan Kautz, and Gabriel Brostow. Hierarchical subquery evaluation for active learning on a graph. In *CVPR*, 2014.

[101] David JC MacKay and David JC Mac Kay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

[102] Babak Mahdian and Stanislav Saic. Detection of copy–move forgery using a method based on blur moment invariants. *Forensic science international*, 171(2):180–189, 2007.

[103] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497–1503, 2009.

[104] Tomasz Malisiewicz and Alexei A Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007.

[105] VT Manu and BM Mehtre. Visual artifacts based image splicing detection in uncompressed images. In *IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)*, 2015.

[106] Justin T Maxfield, Westri D Stalder, and Gregory J Zelinsky. Effects of target typicality on categorical search. *Journal of vision*, 14(12):1–1, 2014.

[107] Tajuddin Manhar Mohammed, Jason Bunk, Lakshmanan Nataraj, Jawadul H Bappy, Arjuna Flenner, BS Manjunath, Shivkumar Chandrasekaran, Amit K Roy-Chowdhury, and Lawrence Peterson. Boosting image forgery detection using resampling detection and copy-move analysis. 2018.

[108] Rodrigo Moraes, Joao Francisco Valiati, and Wilson P GaviãO Neto. Document-level sentiment classification: An empirical comparison between svm and ann. *Expert Systems with Applications*, 40(2):621–633, 2013.

[109] Ghulam Muhammad, Munner H Al-Hammadi, Muhammad Hussain, and George Bebis. Image forgery detection using steerable pyramid transform and local binary pattern. *Machine Vision and Applications*, 25(4):985–995, 2014.

[110] Ghulam Muhammad, Muhammad Hussain, and George Bebis. Passive copy move image forgery detection using undecimated dyadic wavelet transform. *Digital Investigation*, 9(1):49–57, 2012.

[111] L. Nataraj, A. Sarkar, and B. S. Manjunath. Improving re-sampling detection by adding noise. In *SPIE, Media Forensics and Security*, volume 7541, 2010.

[112] Lakshmanan Nataraj, Anindya Sarkar, and Bangalore S Manjunath. Adding gaussian noise to "denoise" jpeg for detecting image resizing. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1493–1496. IEEE, 2009.

[113] Lakshmanan Nataraj, Anindya Sarkar, and Bangalore S Manjunath. Improving re-sampling detection by adding noise. In *Media Forensics and Security II*, volume 7541, page 75410I. International Society for Optics and Photonics, 2010.

[114] Tejaswi Nimmagadda and Anima Anandkumar. Multi-object classification and unsupervised scene understanding using deep learning features and latent tree probabilistic models. *arXiv preprint arXiv:1505.00308*, 2015.

[115] Parham Noorzad, Michelle Effros, and Michael Langberg. The unbounded benefit of encoder cooperation for the k-user mac. *IEEE Transactions on Information Theory*, 64(5):3655–3678, 2018.

[116] Sangmin Oh, Anthony Hoogs, Amitha Perera, Naresh Cuntoor, Chia-Chih Chen, Jong Taek Lee, Saurajit Mukherjee, JK Aggarwal, Hyungtae Lee, Larry Davis, et al. A large-scale benchmark dataset for event recognition in surveillance video. In *IEEE conference on Computer vision and pattern recognition (CVPR)*, pages 3153–3160, 2011.

[117] Francisco Javier Ordóñez and Daniel Roggen. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors*, 16(1):115, 2016.

[118] Sujoy Paul, Jawadul H. Bappy, and Amit Roy-Chowdhury. Non-uniform subset selection for active learning in structured data. In *CVPR*, 2017.

[119] Pedro HO Pinheiro and Ronan Collobert. Recurrent convolutional neural networks for scene labeling. In *International Conference on Machine Learning*, 2014.

[120] Pedro O Pinheiro, Tsung-Yi Lin, Ronan Collobert, and Piotr Dollár. Learning to refine object segments. In *European Conference on Computer Vision*, pages 75–91. Springer, 2016.

[121] Oluwatoyin P Popoola and Kejun Wang. Video-based abnormal human behavior recognition—a review. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(6):865–878, 2012.

[122] Ronald Poppe. A survey on vision-based human action recognition. *Image and vision computing*, 28(6):976–990, 2010.

[123] Chi-Man Pun, Xiao-Chen Yuan, and Xiu-Li Bi. Image forgery detection using adaptive oversegmentation and feature point matching. *IEEE Transactions on Information Forensics and Security*, 10(8):1705–1716, 2015.

[124] Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan. Deep learning for steganalysis via convolutional neural networks. In *SPIE/IS&T Electronic Imaging*, 2015.

[125] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *CVPR*, 2009.

[126] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie. Objects in context. In *ICCV*, 2007.

[127] Yuan Rao and Jiangqun Ni. A deep learning approach to detection of splicing and copy-move forgeries in images. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.

[128] Marcus Rohrbach, Sikandar Amin, Mykhaylo Andriluka, and Bernt Schiele. A database for fine grained activity detection of cooking activities. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1194–1201. IEEE, 2012.

[129] Mehrsan Javan Roshtkhari and Martin D Levine. Online dominant and anomalous behavior detection in videos. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2611–2618. IEEE, 2013.

[130] Seung Ryu, Matthias Kirchner, Min Lee, and Heung Lee. Rotation invariant localization of duplicated image regions based on zernike moments. *IEEE Transactions on Information Forensics and Security*, 8(8):1355–1370, 2013.

[131] Seung-Jin Ryu and Heung-Kyu Lee. Estimation of linear transformation by analyzing the periodicity of interpolation. *Pattern Recognition Letters*, 36:89–99, 2014.

[132] Mohammad Sabokrou, Mahmood Fathy, Mojtaba Hoseini, and Reinhard Klette. Real-time anomaly detection and localization in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 56–62, 2015.

[133] Babak Saleh, Ahmed Elgammal, and Jacob Feldman. The role of typicality in object classification: Improving the generalization capacity of convolutional neural networks. *arXiv preprint arXiv:1602.02865*, 2016.

[134] Venkatesh Saligrama and Zhu Chen. Video anomaly detection based on local statistical aggregates. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2112–2119. IEEE, 2012.

[135] Anindya Sarkar, Lakshmanan Nataraj, and Bangalore S Manjunath. Detection of seam carving and localization of seam insertions in digital images. In *Proceedings of the 11th ACM workshop on Multimedia and security*, 2009.

[136] Mark Schmidt. Ugm: A matlab toolbox for probabilistic undirected graphical models, 2010.

[137] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.

[138] Burr Settles. Active learning literature survey. *University of Wisconsin, Madison*, 52(55-66), 2010.

[139] Burr Settles. Active learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 6(1):1–114, 2012.

[140] Lixin Shi, Yuhang Zhao, and Jie Tang. Batch mode active learning for networked data. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2):33, 2012.

[141] Dijana Tralic, Ivan Zupancic, Sonja Grgic, and Mislav Grgic. Comofod—new database for copy-move forgery detection. In *ELMAR, 2013 55th international symposium*, pages 49–54. IEEE, 2013.

[142] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015.

[143] Luisa Verdoliva, Davide Cozzolino, and Giovanni Poggi. A feature-based approach for image tampering detection and localization. In *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014.

[144] Sudheendra Vijayanarasimhan and Kristen Grauman. Large-scale live active learning: Training object detectors with crawled data and crowds. *IJCV*, 108(1-2):97–114, 2014.

[145] Francesco Visin, Marco Ciccone, Adriana Romero, Kyle Kastner, Kyunghyun Cho, Yoshua Bengio, Matteo Matteucci, and Aaron Courville. Reseg: A recurrent neural network-based model for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2016.

[146] Julia Vogel and Bernt Schiele. A semantic typicality measure for natural scene categorization. In *Joint Pattern Recognition Symposium*, pages 195–203. Springer, 2004.

[147] Carl Vondrick and Deva Ramanan. Video annotation and tracking with active learning. In *NIPS*, 2011.

[148] B. Wang, D. Lin, H. Xiong, and Y.F. Zheng. Joint inference of objects and scenes with efficient learning of text-object-scene relations. *Multimedia, IEEE Transactions on*, PP(99):1–1, 2016.

[149] Wei Wang, Jing Dong, and Tieniu Tan. Exploring dct coefficient quantization effects for local tampering detection. *IEEE Transactions on Information Forensics and Security*, 9(10):1653–1666, 2014.

[150] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage - a novel database for copy-move forgery detection. In *IEEE International Conference on Image processing (ICIP)*, 2016.

[151] Christian Wojek and Bernt Schiele. A dynamic conditional random field model for joint labeling of object and scene classes. In *ECCV*. 2008.

[152] Qiong Wu, Shao Sun, Wei Zhu, Guo Li, and Dan Tu. Detection of digital doctoring in exemplar-based inpainted images. In *International Conference on Machine Learning and Cybernetics*, 2008.

[153] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010.

[154] Tan Xiao, Chao Zhang, and Hongbin Zha. Learning to detect anomalies in surveillance video. *IEEE Signal Processing Letters*, 22(9):1477–1481, 2015.

[155] Dan Xu, Rui Song, Xinyu Wu, Nannan Li, Wei Feng, and Huihuan Qian. Video anomaly detection based on a hierarchical activity discovery within spatio-temporal contexts. *Neurocomputing*, 143:144–152, 2014.

[156] Jian Yao, Sanja Fidler, and Raquel Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012.

[157] Jonathan S Yedidia, William T Freeman, and Yair Weiss. Constructing free-energy approximations and generalized belief propagation algorithms. *Information Theory, IEEE Transactions on*, 51(7):2282–2312, 2005.

[158] Jun Yue, Zhenbo Li, Lu Liu, and Zetian Fu. Content-based image retrieval using color and texture fused features. *Mathematical and Computer Modelling*, 54(3):1121–1127, 2011.

[159] Ming Zeng, Le T Nguyen, Bo Yu, Ole J Mengshoel, Jiang Zhu, Pang Wu, and Joy Zhang. Convolutional neural networks for human activity recognition using mobile sensors. In *International Conference on Mobile Computing, Applications and Services (MobiCASE)*, pages 197–205, 2014.

[160] Lei Zhang, Xiantong Zhen, and Ling Shao. Learning object-to-class kernels for scene classification. *TIP*, 23(8):3241–3253, 2014.

[161] Yimeng Zhang, Xiaoming Liu, Ming-Ching Chang, Weina Ge, and Tsuhan Chen. Spatio-temporal phrases for activity recognition. In *ECCV*. 2012.

[162] Ying Zhang, Huchuan Lu, Lihe Zhang, Xiang Ruan, and Shun Sakai. Video anomaly detection based on locality sensitive hashing filters. *Pattern Recognition*, 59:302–311, 2016.

[163] Ying Zhang, Lei Lei Win, Jonathan Goh, and Vrizlynn LL Thing. Image region forgery detection: A deep learning approach. In *Proceedings of the Singapore Cyber-Security Conference (SG-CRC)*, 2016.

[164] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *IEEE International Conference on Computer Vision*, 2015.

[165] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014.

[166] Yingying Zhu, Nandita M Nayak, and Amit K Roy-Chowdhury. Context-aware activity recognition and anomaly detection in video. *IEEE Journal of Selected Topics in Signal Processing*, 7(1):91–101, 2013.

[167] Yingying Zhu, N.M. Nayak, and A.K. Roy-Chowdhury. Context-aware activity modeling using hierarchical conditional random fields. *PAMI*, 37(7):1360–1372, 2015.