CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS FOR PEDESTRIAN DETECTION

by

VIVEK ARVIND BALAJI

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2016

To my parents and my brother.

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervising professor, Dr. Junzhou Huang without whom this thesis would not have been possible. His immense encouragement and supervision are the main reasons of the successful outcomes of my research. I sincerely express my gratitude to Dr. Jeff (Yu) Lei and Dr. Jia Rao for serving on my thesis committee.

I would like to thank Zheng Xu, Jiawen Yao, Sheng Wang, Ashwin Raju and other friends in my lab for their motivation. I would also like to thank my friends Viswanathan Kavassery Rajalingam, Deepakraj Srinivasan, Eshwar Ravindran, Sreerathan Chadalavada, Vivek Sundararajan and Srinivas Varadharajan for their constant support and encouragement.

November 18, 2016

ABSTRACT

CONVOLUTIONAL AND RECURRENT NEURAL NETWORKS FOR PEDESTRIAN DETECTION

VIVEK ARVIND BALAJI, M.S.

The University of Texas at Arlington, 2016

Supervising Professor: Dr. Junzhou Huang

Pedestrian Detection in real time has become an interesting and a challenging problem lately. With the advent of autonomous vehicles and intelligent traffic monitoring systems, more time and money are being invested into detecting and locating pedestrians for their safety and towards achieving complete autonomy in vehicles. For the task of pedestrian detection, Convolutional Neural Networks (ConvNets) have been very promising over the past decade. ConvNets have a typical feed-forward structure and they share many properties with the visual system of the human brain. On the other hand, Recurrent Neural Networks (RNNs) are emerging as an important technique for image based detection problems and they are more closely related to the visual system due to their recurrent connections. Detecting pedestrians in a real time environment is a task where sequence is very important and it is intriguing to see how ConvNets and RNNs handle this task. This thesis hopes to make a detailed comparison between ConvNets and RNNs for pedestrian detection, how both these techniques perform on sequential pedestrian data, their scopes of research and what are their advantages and disadvantages. The comparison

is done on two benchmark datasets - TUD-Brussels and ETH Pedestrian Datasets and a comprehensive evaluation is presented to see how research on these topics can be taken forward.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 Pedestrian Detection

Machines have become an integral part of our everyday lives and the interaction between us and machines is becoming vital as time passes by. Artificial Intelligence involves machines getting man-like abilities, sometimes more. Recent advancements in technology have made machines to interact with human beings in a more sophisticated way but there is a long way to go before complete artificial intelligence is achieved. Therefore, there is a necessity for machines to detect and recognize objects that a human would recognize, including other human beings. Object detection is something that scientists have been teaching machines to perfect over the past few decades. Detecting people, especially pedestrians is one critical sub problem in object detection and it is a separate area of research in its own, which has grown popular over the years due its varied applications.

Pedestrian Detection is being used in a variety of fields such as Autonomous Driving, Traffic Monitoring Systems, Intelligent Surveillance, etc. There is a lot of research being done on detecting pedestrians through machine vision approaches, and this task is challenging due to a variety of reasons like occlusions in pedestrian data, fast moving pedestrians, jaywalking pedestrians, etc. A lot of questions like "Do the state-of-the-art methods work well?","Is it computationally burdening","What is the best approach for detecting pedestrians?", "Are Deep Neural Networks the correct way to proceed?", "What are the potential problems that are likely to arise?", do not have a concrete answer yet. This thesis aims to find some answers by doing

a comparison between two specific Deep Learning approaches Convolutional Neural Networks (CNNs or ConvNets) and Recurrent Neural Networks (RNNs), how they perform on this particular task of pedestrian detection and what are the potential areas of improvement.

1.2 Convolutional Neural Networks (ConvNets)

Convolutional Neural Networks (CNNs or ConvNets) are a type of neural networks that has been proven very effective in a variety of applications such as object detection & recognition, image classification and several Natural Language Processing (NLP) applications. ConvNets have been successful in recognizing faces, people, traffic signs, and have also been powering self driving cars, robots, etc.

ConvNets are nothing but the biologically inspired version of Multi Layer Perceptrons. The motivation comes from the organization of visual cortex in animals. The visual cortex in animals has an extremely complex structure and the neurons in the visual cortex respond to a restricted portion of the entire visual field in front of them. These restricted portions are called as receptive fields. These receptive fields of the individual neurons add together to form the whole visual field. This response to particular stimuli can be mathematically represented as a convolution operation and this can be used to build a neural network which would replicate the same organizational structure of animal visual cortex, hence the name Convolutional Neural Networks. Below is a detailed explanation of how this can be done.

There are four main operations that are critical to a typical ConvNet, namely Convolution, Non-linearity, Pooling or Sub sampling and Classification. These operations are achieved using three fundamental layers in the network architecture. They are the Convolutional Layer, the Pooling Layer and the Fully Connected layer. For example, LeNet5 [20] is one of the classic ConvNet architectures which can be used

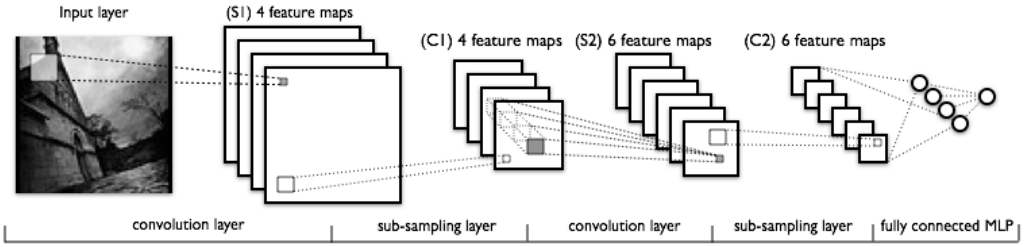

Figure 1.1: LeNet5 Architecture

to understand how these layers are put to use. The Lenet architecture is shown in Figure 1.1.

Intuitively, the Convolutional Layer does the convolution operation which is nothing but extracting features from the input image, called as feature maps. This operation preserves the spatial information in the image pixels by learning features using filters. A filter or a kernel in ConvNets is like a small window or a matrix that is slided over the input image. During this sliding process, the filters compute the dot product of the corresponding image pixels and produce a feature map or an activation map. The filters act as feature detectors from the input image and different values of feature matrices will produce different feature maps.

The pooling layer (or the sub-sampling layer) reduces the dimensionality of the generated feature maps. This can be done in several ways. Consider a window on the feature map. One way is to sum all the values of the feature map in the window. Other ways are to take the max (called Max Pooling) or the average (called Average Pooling) of the values of the feature that come in the window. This way it can be ensured that the most important information in the feature maps is retained.

The fully connected layer is nothing but a traditional Multi-Layer Perceptron with an activation function like Softmax, SVM, etc. The name comes from the fact that every neuron in this layer is connected to every other neuron in the previous layer. This layer in most of the cases does the classification part. The output from

the convolutional and the pooling layers represent very high level features from the input image and this layer uses those features for classifying categories specified in the training data. ConvNets have been most useful in image based applications due to their effective representation of image data and the ability to learn image features with great accuracy and precision. The following subsection describes another type of neural networks called the Recurrent Neural Networks (RNNs) which is considered equally promising.

1.3 Recurrent Neural Networks (RNNs)

Traditional neural networks do not have the ability to remember information from the past, which seemed to be a major shortcoming until the advent of Recurrent Neural Networks (RNNs). Recurrent Neural Network is another type of artificial neural networks which have loops in them, which allow information to stay. RNNs can be conceived as a normal neural network with a chain like structure where there are multiple copies of a network and each network passes information to its successor and so on. RNNs have got increased attention in the recent past and incredible results have been achieved using RNNs in NLP applications such as speech recognition[7], handwriting recognition[8], language translation, language modelling[9] and sometimes also in object detection related applications. RNNs are expected to perform well in image based applications based on the assumptions that they might be able to look up past information (say in a past video frame) and use that information in the present frame.

However, the problem of long-term dependencies would persist in traditional RNNs. When it comes to long-term memories, RNNs cannot remember information from the long past. When the gap between the relevant information and the point where it is needed increases, RNNs cannot remember the connection. This

Figure 1.2: Sample annotations on ETH & TUD-Brussels datasets

has been explored in detail in [15]. This is where LSTMs come handy. Long Short Term Memory networks (LSTMs) do not have this long term dependency problem. LSTMs, which were introduced in [16], are explicitly designed to overlook this long time dependency problem. Like RNNs, LSTMs also have chain like structure but the difference is that there are cell states in the LSTM cells. The states of these cells can be modified by the use of gates. Gates are structures that optionally let information pass through to other LSTM cells. Gates control whether whole or part of the information should be passed to other cells. This makes it possible for LSTMs to control which information should be retrieved from the past, so as to make effective detection.

1.4 Goals of this thesis

There has been a reasonable amount of research done in image based detection in RNNs [17,18,19]. However, application of RNNs or LSTMs to pedestrian detection has not been explored much yet. One of the primary goals of this thesis is to

present a comparison study of ConvNets versus RNNs for pedestrian detection and to evaluate their performance on this particular task. The comparison is done on two datasets namely the TUD_Brussels dataset, which contains about 1500 frames with 2000 labelled pedestrians and the ETH pedestrian dataset which contains about 3400 frames with 17800 labelled pedestrians. Sample annotations of both datasets are shown in Figure 1.2. This thesis also aims to present directions of research in future, how pedestrian detection approaches can be taken forward to achieve effective results.

CHAPTER 2

CONVOLUTIONAL NEURAL NETWORKS FOR PEDESTRIAN DETECTION

2.1 ConvNets for Pedestrians: A Perspective

In pedestrian detection, there has been an upsurge in the use of ConvNets over the last decade [1,2,3,4,5], due to their success in similar image analysis tasks. The fundamental idea behind ConvNets is to learn complex features from pixel-level contents, capitalizing on a sequence of operations such as filtering, normalization, pooling, etc. For example, the features proposed by Viola & Jones [6] mainly come from the assumption that the shape of a pedestrian is recognized by abrupt changes in pixel intensity in the contour regions of the body. In a broad sense, the pipeline of ConvNets in pedestrian detection is very much in line with that of object detection i.e., the pipeline starts from raw image from which the proposal of candidate regions are proposed, then higher level representations are extracted to be applied pixel-to-pixel. Then these extracted features are fed to a classifier which estimates if the extracted region depicts a pedestrian. This common pipeline is explained in detail in [21] as shown in Figure 2.1.

Going down the history, the two important works in pedestrian detection are the HOG detector [22] and the Viola-Jones detector [6]. The Viola-Jones detector was improved by Dollar et al., [23] which proposed the Integral Channel Features involving several types of features such as LUV channels, grayscale features, etc. those can be determined by integral images. On the other hand, HOG features gave rise to many other state of the art pedestrian detection methods including the Deformable Parts Model (DPM) [24] which was a major breakthrough. These early pedestrian detection

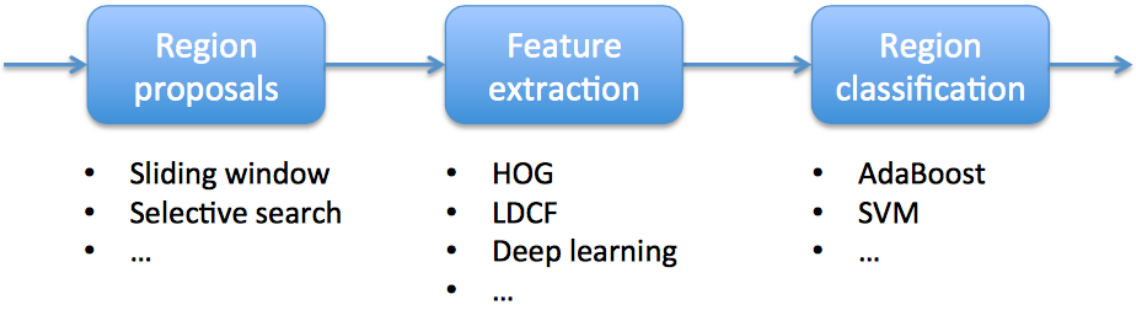

Figure 2.1: A common pipeline for pedestrian detection

models were more like hand crafted feature extractors, but the Deep Convolutional Neural Network (DCNN) Architecture (popularly known as AlexNet) proposed by Krizhevsky et al. [25] achieved record-breaking results in ImageNet Large Scale Visual Recognition Challenge 2012. The AlexNet architecture is shown in Figure 2.2.

Pedestrian detection has seen a bunch of deep learning based approaches proposed in the recent past. Sermanet et al., [26] proposed a two-layer ConvNet model and [27] proposes a joint deep learning based ConvNet framework which takes important components for pedestrian detection such as feature extraction, deformation model, person-to-person relation and occlusion model jointly into account. Other approaches like [28] inputs each layer with contextual features extracted around the candidate regions. It is important to note that in [26], each layer is first initialized with convolutional sparse coding and then the whole ConvNet is fine tuned for pedestrian detection. Then the features from the last layer are used for detection. This approach learns to predict from YUV input while other models use hand crafted features.

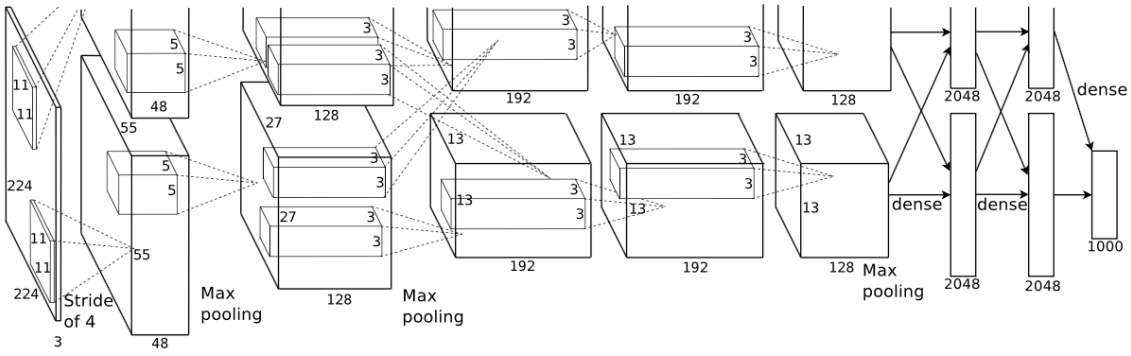

Figure 2.2: AlexNet Architecture

2.2 ConvNets for Objects

A few main reasons why ConvNets are used for image based recognition are (i) Feature extraction and classification are integrated into a single structure and they are totally adaptive, (ii) ConvNets extract 2D image features at high scales and (iii) It is resistant to geometric and local distortions in an image. ConvNets largely rely on huge training datasets, training procedure based on backpropagation with optimization algorithms like gradient descent, etc. In the context of detecting pedestrians, the last layer typically contains a single neuron that acts as a classifier that determines if the input region contains a pedestrian or not. In spite of these models being powerful, they need large datasets with annotations to yield accurate results.

As mentioned earlier, pedestrian detection shares some of its approaches and methodologies with those of object detection. So this yields a whole lot of possibilities of applying some of the best object detection approaches to this particular task of pedestrian detection. For instance, approaches like GoogLeNet [29] proposed by C. Szegedy et al., which won the ImageNet Large-Scale Visual Recognition Challenge in 2014 (ILSVRC2014) in Object Detection; VGG [30] proposed by Simoyan et al., which won the same challenge (ILSVRC2014) in Classification & Localization; AlexNet [25]

which won the ILSVRC2012 challenge; OverFeat [5] proposed by Sermanet et al., which won the ILSVRC2013 challenge in localization for the ImageNet and approaches like Region Based Convolutional Neural Network (R-CNN) [31] for the Pascal VOC categories, etc. can be applied to pedestrian detection task.

2.3 Overfeat for Pedestrian Detection

This thesis takes the object recognition approach Overfeat [5] into comparison and evaluates how well it performs for the task of pedestrian detection against RNNs. This paper presents three fundamental ideas for handling objects of various sizes, shapes and positions in an image,

1. To apply ConvNets in various portions of the image, at various scales in a sliding window fashion. In spite of this, there may be a perfectly identifiable portion of an object, but not the whole object which leads to a reasonable classification but poor detection and localization.
2. To train the system to produce a prediction of location & size of bounding boxes corresponding to an object relative to the window, in addition to a distribution over categories for each window.
3. To accumulate the evidence for each category at every location and size.

This paper explores three vision tasks namely classification, localization and detection, in the increasing order of difficulty while each task being the sub task of the next one.

2.3.1 Classification

In the classification task, each image is assigned a label pertaining to the main object in the image. The architecture is similar to that of AlexNet (Figure 2.2) - it has eight layers. The first five are convolutional layers and the remaining are the fully

connected layers. During training, the model uses fixed sized inputs like AlexNet and the output of the last fully connected layer is fed to a thousand way softmax. In spite of the architecture being similar to that of AlexNet, there are some mentionable differences between the two. They are

1. The input images are not normalized for their contrast - to reduce glare effects.
2. Pooling regions do not overlap
3. Smaller stride for improved accuracy.

Each image is down-sampled and then 5 random crops of size 221x221 are extracted. The weights are randomly initialized and then updated using Stochastic Gradient Descent (SGD). Then a dropout of 0.5 is employed on the fully connected layers. Finally, while testing, the image is entirely explored by running the network densely at different locations and scales.

2.3.2 Localization

After the classification, the bounding box of each classified object is returned along with a confidence value with respect to the ground truth. This confidence value (also called as Intersection over Union) must be larger than 0.5 for the bounding box to qualify. For localization, the previous model is modified by replacing the classifier with a regression network. Then the regression predictions are combined with the classification results in all the locations. The regression network is trained with an L2 loss after which the individual predictions are merged together according a greedy strategy.

2.3.3 Detection

Detection training is very much similar to the classification training except that this is done in a spatial manner. One main advantage in the detection part is that

multiple locations of the image can be trained simultaneously and the weights can be shared between these locations. The primary difference between the localization and the detection tasks is the necessity to predict a background class when no object is present. A challenge is that the traditional way of bootstrapping the negative samples makes the training complicated and may cause mismatches. This approach prevents such problems by selected a few interesting negative examples per image. This may be computationally expensive but makes the training process smoother. This gives a more granular idea of how it can be applied to the pedestrian detection task, the results of which are mentioned in Chapter 4.

CHAPTER 3

RECURRENT NEURAL NETWORKS FOR PEDESTRIAN DETECTION

3.1 Recurrent Neural Networks: A Perspective

Recurrent Neural Networks is a type of artificial neural networks that have loops in them. The connections between the units of an RNN form a cycle and thus are mostly used to handle sequential information. Traditional neural networks such as the ConvNets have the limitation of not remembering the information from the past. They work mainly on the assumption that the inputs from the previous stages are totally independent to each other and to the data in the next stage. RNNs overcome this problem by performing the same task for every element of a sequence (say a sequence of images), with the output being dependent on the previous computations. This enables the information to persist, allowing the network to have some sort of memory while dealing with sequential data. Figure 3.1 shows a simple RNN A with an input x_t and an output h_t. The important thing to be noticed is that, there is a loop in the network A which allows information to be passed from one stage of the network to the other. This might seem a little unclear but if closely observed, RNNs can be conceived as multiple copies of the same network, while information is being transferred from one copy to the other. It is like unrolling a single RNN into several copies of simple neural networks, as shown in Figure 3.2.

3.2 RNNs for Pedestrian Detection

In terms of pedestrian detection or object detection, RNNs as such would not be completely useful. Although RNNs have the advantage of remembering informa-

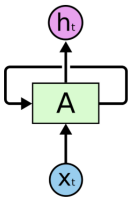

Figure 3.1: A simple RNN

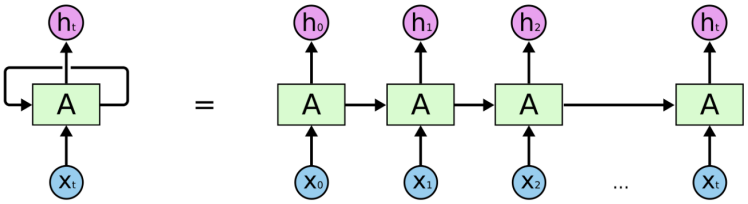

Figure 3.2: RNN Unrolled

tion from the past, some applications would require RNNs to control the information that should be remembered. In other words, employing RNNs to Image based detection/recognition might require that the RNN forget a part of the information from the past and remember only certain past information. For example, if an RNN is required to remember a pedestrian from 100th and the 500th frame of an image sequence, it must be able to discard all the other irrelevant information from the past. While a basic RNN architecture can handle this type of information, if the gap between two required pieces of information increases, a simple RNN architecture would fail. This problem was solved by Hochreiter et al., in [16], where they proposed a new type of RNN called the Long Short Term Memory (LSTM) network. The LSTMs are capable of handling the long term dependency problem, by the use of state information and structures called gates, which can be used to decide which information should be retained and where the retained information should be used.

With respect to Pedestrian Detection and related applications such as object detection, human trajectory detection, etc., RNNs have been extensively used in the

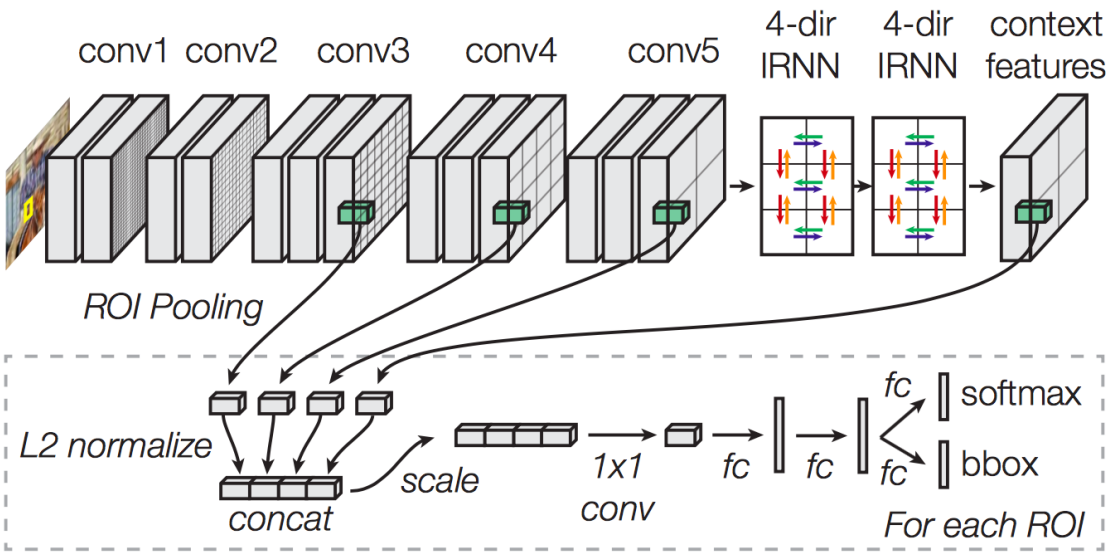

Figure 3.3: Inside-Outside Net (ION)

recent past in [10,11,12,13,32,14] but not enough. There is still a lot of scope for usage and improvement of RNNs in these image based applications and the neural network community is moving towards RNNs for achieving better results. [10] proposes Inside-Outside Net (ION) shown in Figure 3.3, an object detector that considers information both inside and outside the regions of interest. Another important aspect of this paper is the use of skip pooling in the architecture to extract VGG [30] features at multiple levels of abstraction. On top of this a 2x stacked 4-directional IRNNs [33] are used to extract context features describing the image. The extracted features are then used to evaluate over 2000 regions of interest which were proposed during the feature extraction. Then, during each proposal a fixed sized, L2-normalized, concatenated and scaled descriptor is extracted. Finally, the extracted descriptors are processed by two fully connected layers to produce predictions and bounding boxes over the objects.

3.2.1 RNNs with ConvNets for Pedestrian Detection

There also have been other approaches which incorporate recurrent architecture into convolutional layers for static object detection. One such approach is defined in [32] where the authors define a new layer called Recurrent Convolutional Layer (RCL) interleaved with max pooling layers. This network has a regular feed forward Convolutional layer to minimize computational overload and on top of this, lay 4 RCLs with max pooling layers in the middle. Then there is a global max pooling layer which gives a feature vector with image representations. Finally, a softmax classifier classifies the feature vectors. While this could be one of the ways of adding a functionality of the human brain to the ConvNets, there is a possibility that the addition of recurrent connections to ConvNets would not be powerful enough to detect objects in real time scenario.

3.2.2 RNNs for Pedestrian Activity Detection

Other approaches like [11] and [13] take people detection to the next level, where they try to predict the activity of people, right after detecting them. Both the papers address this issue by using several combinations of recurrent deep learning networks and other techniques to predict the activities of people. [11] for instance, uses a combination of a graphical model called Sequential Inference and RNN for activity recognition. After detecting pedestrians in a scene, a classifier is applied on the detections to classify the person's action based on an image window over the detected pedestrian. On top of the individual classifications, an RNN is applied to refine the classification by considering the connections between the actions of people in the scene. This scenario can be represented as a graphical model and the connection between the nodes of the graph are modelled by an RNN architecture. Therefore, in

every refinement the contextual information is updated and at the end a set of refined classification scores are given as output.

On the other hand, [13] studies two problems: Activity Detection & Early Detection. During activity detection, the activity, its start and its end are detected. Whereas, during early detection, the activity and its starting point has to be detected based on very short observations of the activity or a part of the activity. This problem is handled by the use of ranking loss in addition to confidence loss in an LSTM network, to learn models better.

3.3 LSTMs in Pedestrian Detection

In this context, it is more important to model the progression of activities. LSTMs serve this purpose by capturing the necessary observations from the past and used in addition to the current observation to give long range predictions. The ability of LSTMs to remember the spatial relations make them successful in image based detection approaches.

One such approach is the Social-LSTM [12] approach which predicts human trajectory in crowded scenes such as a road. Now, this is a sequential task i.e., to calculate the possible position of a pedestrian with the help of the past position of the pedestrian. However, the motion of pedestrians on the road depends on other people around them. For example, people talking to one another move in the same direction. This approach uses LSTMs to consider this condition and jointly predict their trajectories together based on some social rules, people's mannerisms, etc. Each trajectory in the scene is handled by separate LSTMs and then they are connected together at the end using a "Social Pooling" (S-pooling) layer. This layer only connects LSTMs that are spatially close to each other i.e., the hidden states of LSTMs

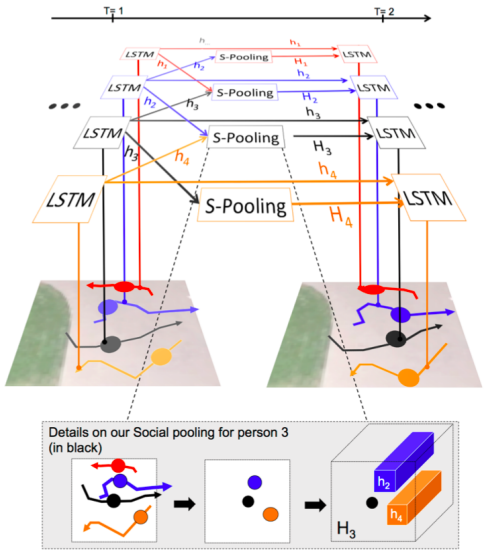

Figure 3.4: Social-LSTM

within a small radius are pooled together. Figure 3.4 shows the architecture of this approach.

3.4 ReInspect: An LSTM based approach for pedestrian detection

This thesis takes the approach called ReInspect [14] into comparison with ConvNets as this would be a very good example for the use of RNNs (LSTMs) for pedestrian detection. This is an end-to-end approach which given an input image, directly outputs a set of bounding boxes over the pedestrians present in the image. This hypothesis generation is achieved using a recurrent LSTM layer with a novel loss function called as the hungarian loss function.

3.4.1 Contributions

These are some novel contributions made by this approach.

1. Aims to show that a stack of LSTMs can be used to decode image representations into real-valued outputs of variable length. This avoids multiple detections on the same person by remembering the past detections.
2. Since it directly outputs prediction, it is unnecessary to generate bounding boxes first, evaluating them with a classifier and then performing some merging or NMS technique on the detections. These techniques do not have direct access to image; rather they work with the generated bounding boxes. This may have repercussions when there are too many pedestrians in the image.
3. The proposed hungarian loss function considers the aspects of localization and detection into account, which leads to more accurate prediction.

3.4.2 Model Overview

Deep Neural Networks are expressive in nature and they have the power of encoding image representations jointly with other instances but they must be augmented in one way or the other for multiple instance prediction. This paper considers Long Short Term Memory (LSTM) networks introduced in [16] to realize this potential. The inherent property of LSTMs makes them capable to tap into higher level representations with ease and to generate variable length outputs.

This approach works more like an encoder-decoder architecture as shown in Figure 3.5. It first encodes the image into high level feature descriptors using the pre-trained weights from GoogLeNet architecture [29] and then decodes the image representations using LSTMs into a set of bounding boxes. The input image is transformed into 1024 dimensional feature vector which contains summarized and rich information about the positions of pedestrians in an image. Then a series of basic LSTM units act as controllers for the decoding part, which produce a new bounding box at each step. This process avoids generation of duplicate bounding boxes using

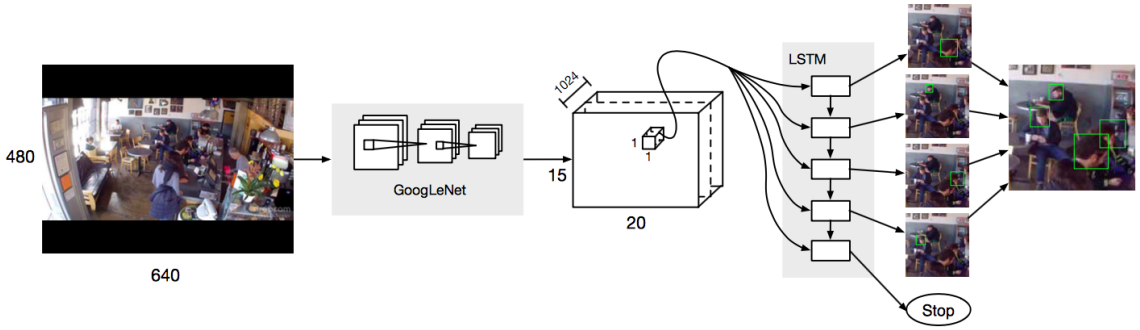

Figure 3.5: ReInspect - Overview

the memory states of the LSTM units. Each bounding box has a confidence score associated with it and the bounding boxes are generated in the order of decreasing confidence. Also, this generation is further narrowed down by specifying a confidence threshold, the bounding boxes below which are ignored by producing a stop symbol. Then all the generations are reconstructed back together as a single image.

For the generation of bounding boxes with confidence greater than the specified threshold and to direct the learning process towards the desired output, a novel loss function called hungarian loss is proposed. Consider the example in Figure 3.6, in which there are 4 hypotheses (green and red boxes) over two ground truth instances (black boxes). There are detection mistakes in this example i.e., poor localization - hypothesis 1, false positive - hypothesis 3, multiple bounding boxes on the same ground truth - hypotheses 2 and 3. The proposed loss function should handle these kinds of mistakes by fine tuning bounding box location (in case of hypothesis 1), ignore false positives by assigning a low confidence score (in case of hypothesis 3) and ignoring the generation of duplicate bounding boxes (in case of hypothesis 2). To do these kinds of manipulations, a new matching function which assigns a unique bounding box to each ground truth instance is proposed. This matching function

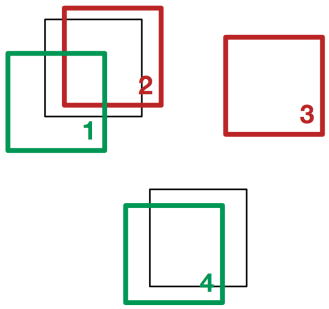

Figure 3.6: Matching ground truths (black) to accepted (green) and rejected (red) candidates

is formulated in such a way that while matching hypothesis to ground truth, three things are considered. They, in the order of priority are,

1. Overlap between the hypothesis and the ground truth,
2. Rank (or) Precedence of the generated hypothesis,
3. Localization - the distance between the bounding box locations.

Thus in Figure 3.6, hypothesis 1 is selected (green box) over hypothesis 2 taking precedence into account and hypothesis 4 is selected (green box) taking localization into account. In a nutshell, this approach address the problem of duplicate bounding box generations and occluded pedestrian detection by decoding image representations into coherent variable length outputs, to achieve an end-to-end approach.

CHAPTER 4

EXPERIMENTS AND RESULTS

4.1 Experimental Setup

The experiments in this thesis were run on an Ubuntu 16.04 LTS machine equipped with an NVIDIA Tesla K40c GPU with 12GB of GPU Accelerator memory, CUDA v7.5 and CuDNN v5. The techniques mentioned in the experiments were written in Python 2.7 and Tensorflow by the respective authors and was reused in Tensorflow 0.10.0 with some modifications accordingly. During training, checkpoint files (*.ckpt) are written out every 10000 iterations along with an event file containing the necessary information to visualize the training process using Tensorboard. During testing, the checkpoint file with optimum hyper-parameters is chosen from the details in the tensorboard and used for detection.

4.2 Datasets

The comparison between ConvNets and RNNs are done on two benchmarks datasets. They are,

1. The TUD-Brussels Dataset - has about 1500 frames of urban driving data with a resolution of 720 x 576 pixels with 2000 fully visible and partially occluded pedestrians labelled in the frames. The annotations are in the Interface Definition Language (*.idl) format having bounding boxes around each pedestrian in the form (x_{min},y_{min},x_{max},y_{max}) denoting the starting and ending (x,y) coordinates of the bounding box.

2. The ETH Pedestrian Dataset - has about 3400 frames of urban pedestrian data collected in different locations and in different climatic conditions. The resolution of each image is 640 x 480 pixels with 17800 pedestrians labelled in each frame. Similar to TUD-Brussels dataset, the annotations of the data are in the Interface Definition Language (*.idl) format having bounding boxes around each pedestrian in the form (x_{min},y_{min},x_{max},y_{max}) denoting the starting and ending (x,y) coordinates of the bounding box.

In both the datasets, the train, test and evaluation data was separated manually in 80 - 20 fashion where 80% of the entire data is used for training and the remaining 20% is used for evaluation and testing combined. Apart from these, the TUD-Brussels dataset has a smaller test dataset called the TUD-Crossing dataset with 200 images of resolutions 640 x 480 pixels. This was also used along with the TUD-Brussels dataset for testing in the experiments.

4.3 Performance Evaluation

As mentioned earlier, two papers (One for ConvNets called as OverFeat [5] and another for RNNs called as ReInspect [14]) are compared in this thesis. This section is organized as follows:

1. Overfeat vs ReInspect - Train & Test: TUD-Brussels Dataset
2. Overfeat vs ReInspect - Train & Test: ETH Pedestrian Dataset

4.3.1 Overfeat vs ReInspect - Train & Test: TUD-Brussels Dataset

4.3.1.1 Receiver Operator Characteristics

The performance of OverFeat and ReInspect in terms of Receiver Operator Characteristic is shown in Figure 4.1. Both the networks were trained with an overall

dropout of 0.5 and with Adam optimizer. The ConvNet approach Overfeat given an AUC measure of 0.72, while the RNN approach ReInspect gives an AUC measure of 0.75.

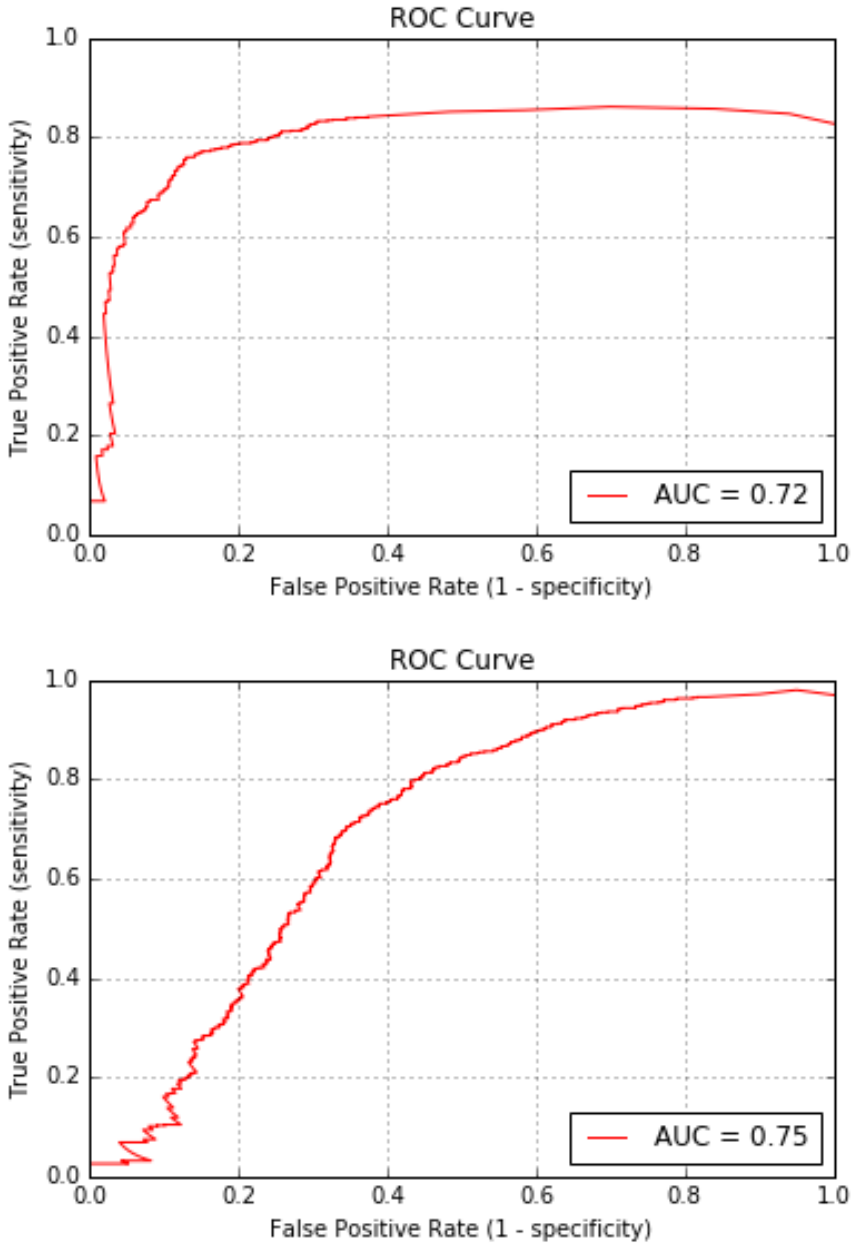

Figure 4.1: ROC Curves of Overfeat (Top) and LSTM (Bottom) on TUD-Brussels Dataset

4.3.1.2 Error Analysis

The performance can also be evaluated by computing the error rate in terms of miss rate and plotting it against the number of False Positives Per Image (FPPI). This evaluation metric was first introduced in [34] and has been in use since then. In this graph, miss rate at 1 FPPI is used as a point of reference for the performance of the approach. Figure 4.2 compares the performance of Overfeat and ReInspect in these terms. It can be seen from the graphs that Overfeat has a miss rate of 20% at 1 FPPI and ReInspect has a miss rate of around 18% at 1 FPPI.

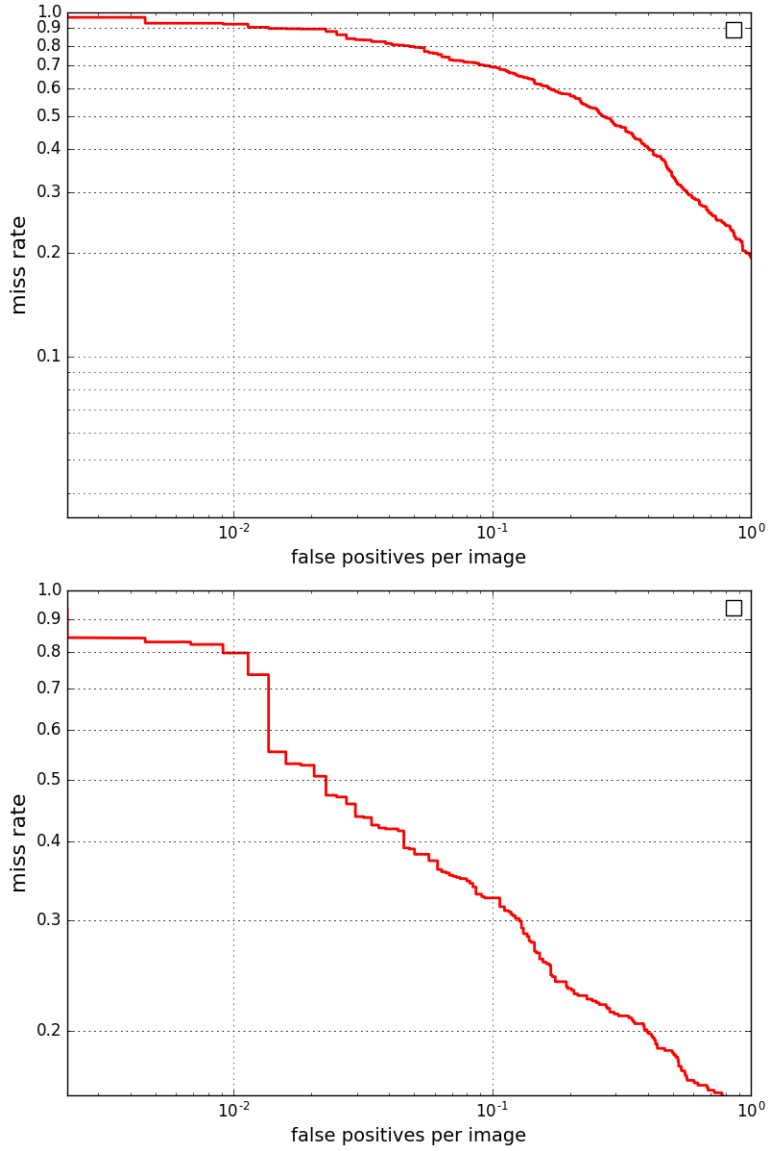

Figure 4.2: Miss Rate vs FPPI curves of Overfeat (Top) and ReInspect (Bottom) on TUD-Brussels Dataset

Some sample detections of both approaches Overfeat and ReInspect on TUD-Brussels test dataset are shown on Figure 4.3 and Figure 4.4 respectively.

Figure 4.3: Sample Detections of Overfeat on TUD-Brussels test dataset

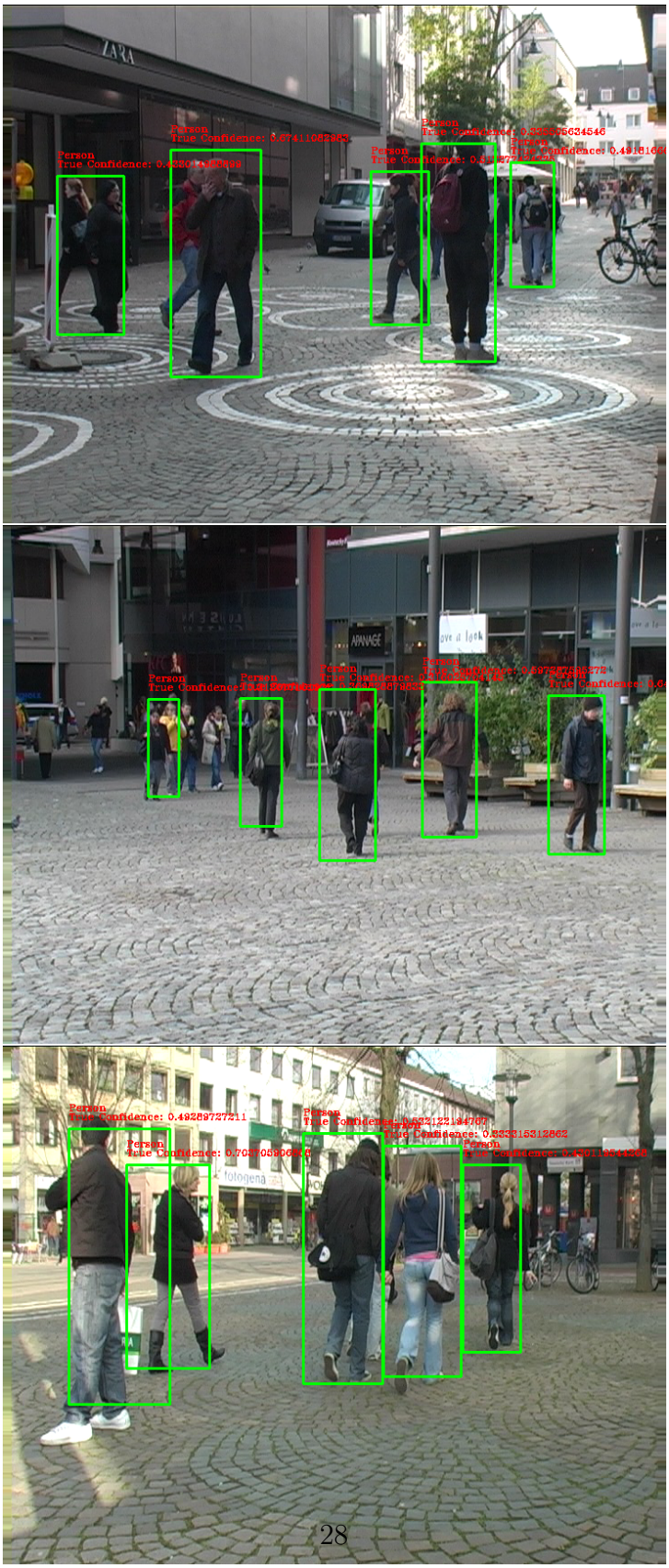

Figure 4.4: Sample Detections of ReInspect on TUD-Brussels test dataset

4.3.2 Overfeat vs ReInspect - Train & Test: ETH Pedestrian Dataset

4.3.2.1 Receiver Operator Characteristics

The performance of Overfeat and ReInspect in terms of Receiver Operator Characteristic is shown in Figure 4.5. In this case, the Overfeat network was trained with an overall dropout of 0.25 and with RMS optimizer while the ReInspect network was trained with an overall dropout of 0.7 and with Adam optimizer as before. The ConvNet approach Overfeat gives an AUC measure of 0.69, while the RNN approach ReInspect gives an AUC measure of 0.80.

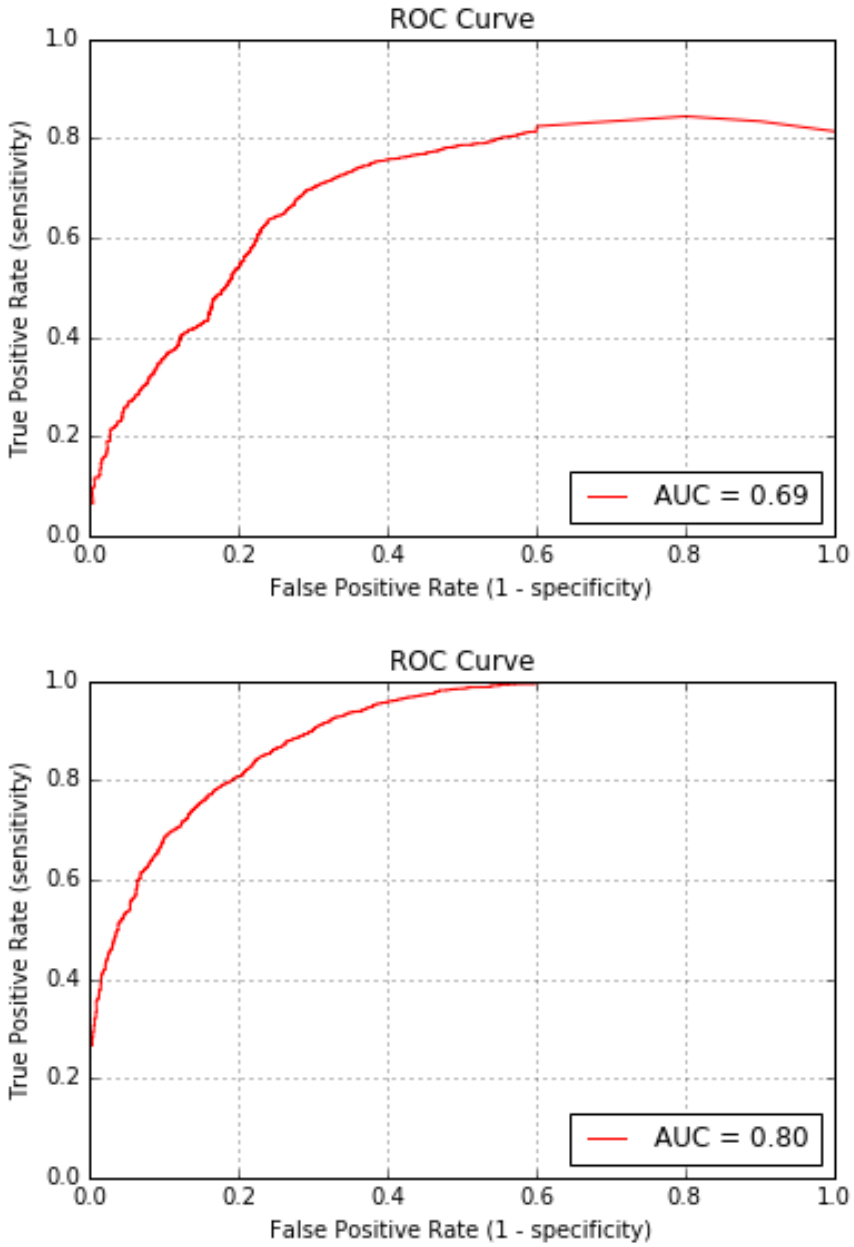

Figure 4.5: ROC Curves of Overfeat (Top) and LSTM (Bottom) on the ETH Pedestrian Dataset

It can be observed from this experiment that the Overfeat Network performs poorly on a mid-sized dataset and it needs more data to learn more features. On the other hand, the ReInspect network performs better with a mid-sized dataset.

4.3.2.2 Error Analysis

The Miss Rate vs FPPI graphs of both the approaches on the ETH Pedestrian Dataset are shown in Figure 4.6. It can be observed from the graphs that Overfeat has a miss rate of 38% at 1 FPPI and ReInspect has a lower miss rate of around 12% at 1 FPPI. The reason behind ReInspect performing better than Overfeat in this case is that the RNN network is robust against the complexity of the data and the number of annotations. ETH Pedestrian Dataset has approximately 18,000 annotated pedestrians. While Overfeat network finds it difficult to keep the miss rate low, ReInspect handles this kind of data better.

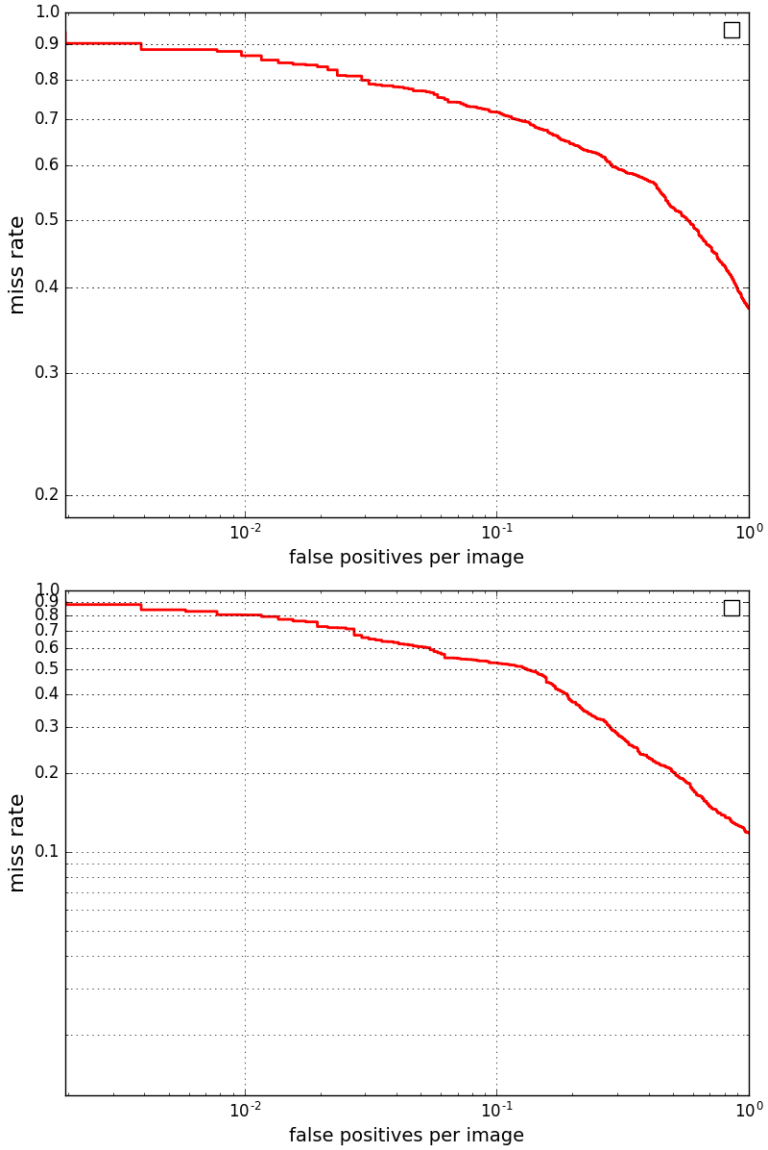

Figure 4.6: Miss Rate vs FPPI curves of Overfeat (Top) and ReInspect (Bottom) on ETH Pedestrian Dataset

Some sample detections of both approaches Overfeat and ReInspect on ETH Pedestrian dataset are shown on Figure 4.7 and Figure 4.8 respectively.

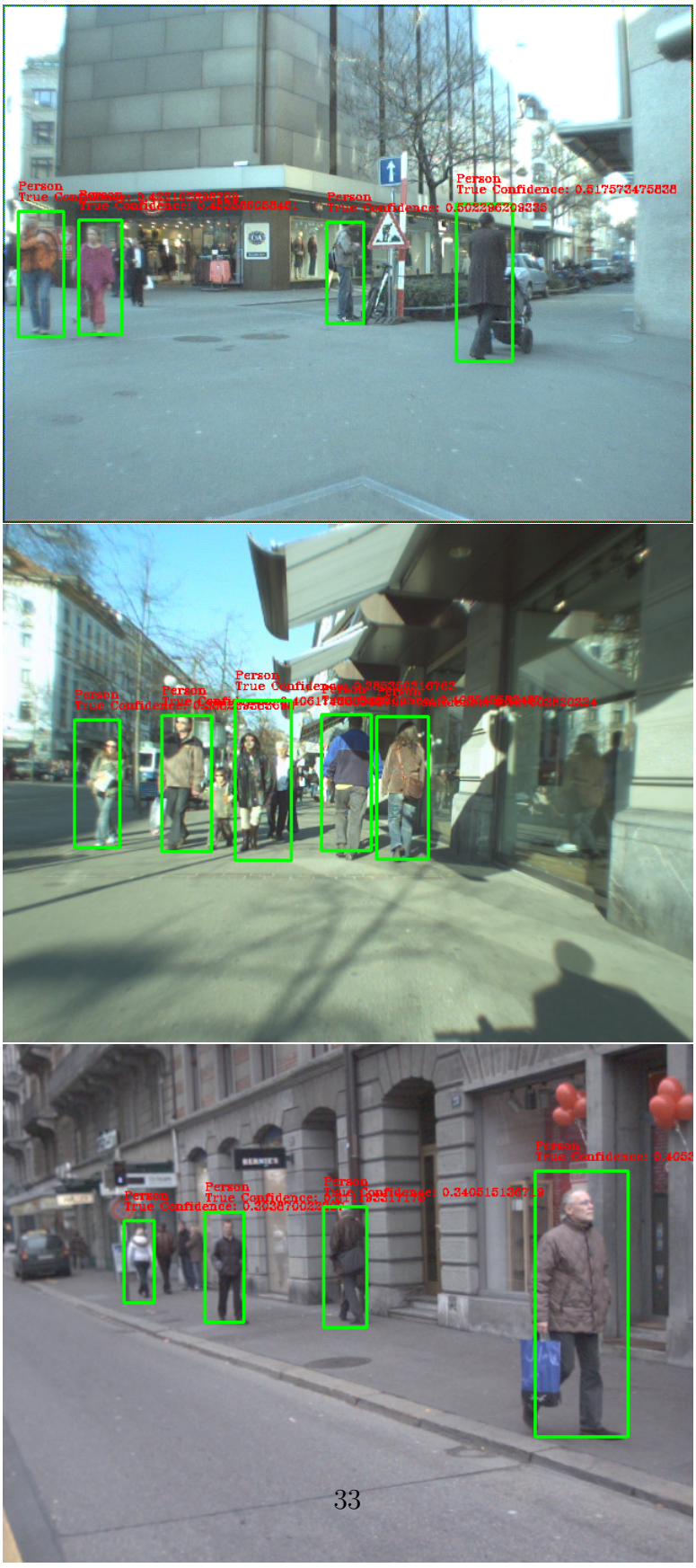

Figure 4.7: Sample Detections of Overfeat on ETH Pedestrian dataset

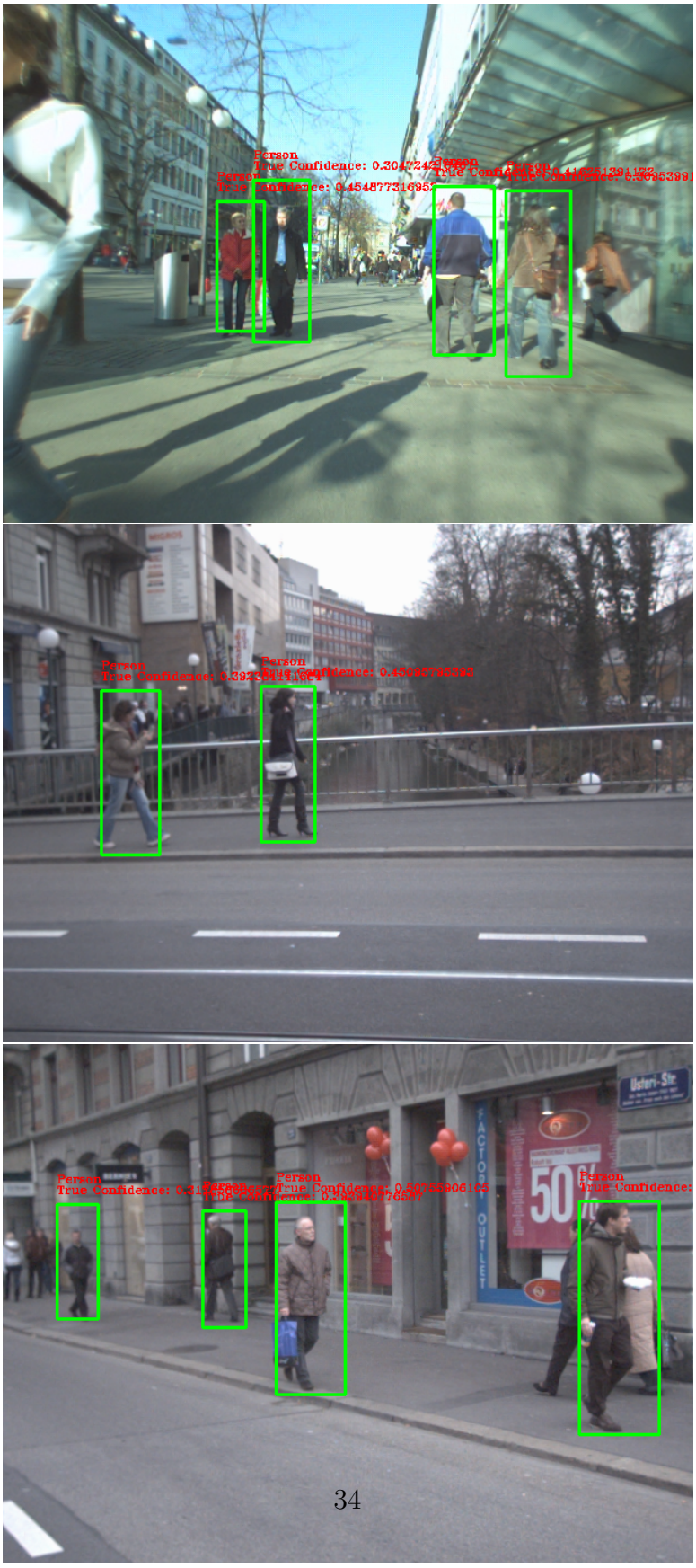

Figure 4.8: Sample Detections of ReInspect on ETH Pedestrian dataset

4.4 Summary of Results

Table 4.1 summarizes the results obtained in the experiments, with other technical details such as the amount of dropout used, the optimizer and the testing durations. It can be observed that though the LSTM based technique "ReInspect" has a better AUC score, it is slower for both the datasets than the ConvNet based technique "Overfeat". This might be because the LSTM technique has more recurrent connections and hence it takes slightly more time. Also, the test times mentioned are for 219 images and 515 images for TUD-Brussels and ETH datasets respectively.

Approach	Dataset	Image Resolution	Optimizer	Dropout	AUC	Miss Rate at 1 FPPI	Test time (sec)
Overfeat	TUD-Brussels	720 x 526	Adam	0.5	0.72	20%	18.46
ReInspect	TUD-Brussels	720 x 526	Adam	0.5	0.75	18%	29.33
Overfeat	ETH	640 x 480	RMS	0.25	0.69	38%	31.72
ReInspect	ETH	640 x 480	Adam	0.7	0.80	12%	51.09

Table 4.1: Summary of Results

4.5 Discussion

The graphs and figures in the last section were the consolidated summary of what evaluations were made for the comparison of ConvNets and RNNs. Section 4.3 gave us many insights about the performance of ConvNets and RNNs for pedestrian detection, their advantages and disadvantages, factors affecting their performance, etc. For instance, RNNs perform better than ConvNets when it is difficult to derive features from the data. ConvNets are intimidated by small/mid-sized data with

complex relations and a large number of annotations. Since RNNs have the ability to remember only the necessary information, they are able to extract more important features from the data. Also in terms of miss rate and false positives, RNNs tend to be robust against changes in information and to be able to handle data of various sizes, features, etc. In real time pedestrian detection, say if an autonomous car wants to drive safely without hurting pedestrians on the road; the data tends to be totally sequential, time-sensitive and stochastic in nature. In such a situation, the approach used for detecting pedestrians must be robust. The experiments in this chapter support the fact that Recurrent Neural Networks might be very promising in the future.

CHAPTER 5

CONCLUSION AND FUTURE WORK

The primary aim of this thesis was to present a comprehensive comparison between Convolutional Neural Networks (ConvNets) and Recurrent Neural Networks (RNNs) for the challenging task of pedestrian detection. The motivation for this thesis is the importance of this task in today's world, the immense potentials of research and the scope of its applications. To be more specific, the whole automotive industry is going behind autonomous cars lately and for such cars, detecting people on the road is a critical task. Deep Neural Networks are expressive in nature, able to learn and extract features from the provided data and has been aiding robots and robot-like structures take intelligent decisions. Particularly, ConvNets have been in use for the past few decades mainly due to their applicability in image data, their effectiveness in representing image data and the ability to learn complex image features. In spite of their effectiveness, pedestrian data is becoming more complex. Pedestrians tend to move in a random, time-sensitive manner and the car that is driving autonomously has to take a lot of decisions in a very short period of time. Sometimes, ConvNets are not able to extract patterns efficiently from this kind of sequential data. On the other hand, Recurrent Neural Networks, which have been primarily used for NLP applications such as Speech Recognition, Handwriting Recognition, etc., can handle this kind of sequential data. In RNNs the data from the previous part of the network is fed into the next stages, thus enabling them to remember information from the past. RNNs have not been explored much in image-based detection applications when compared to ConvNets.

This thesis aims to identify some of the important works that have been done in ConvNets and RNNs for pedestrian detection, to compare them and to provide a direction of research as to what could become effective in detecting pedestrians. From the experiments in the previous chapter, we can come to a conclusion that RNNs are robust to this kind of time-sensitive, sequential data and they could prove to be very assuring. So, research in RNNs would turn out to be fruitful for the task of real time pedestrian detection. The future work would be to take the research forward by exploring the capabilities of RNNs in image based applications and to come up with an approach that would be able to detect pedestrians in an efficient manner. Another direction of further research will be to explore if any new methods like semi-supervised or unsupervised learning could be used for pedestrian detection in future.

REFERENCES

[1] J. Fan et al., "Human Tracking using Convolutional Neural Networks", IEEE Transactions on Neural Networks, 2010.

[2] M. Szarvas et al., "Pedestrian Detection with Convolutional Neural Networks", IEEE Proceedings of Intelligent Vehicles Symposium, 2005.

[3] W. Ouyang et al., "DeepID-Net: multi-stage and deformable deep convolutional neural networks for object detection", CVPR 2014.

[4] W. Ouyang et al., "DeepID-Net: Deformable Deep Convolutional Neural Networks for Object Detection", CVPR 2015.

[5] P. Sermanet et al., "OverFeat: Integrated Recognition, Localization and Detection using Convolutional Networks", CVPR 2014.

[6] Viola et al., "Detecting pedestrians using patterns of motion and appearance", Proceedings of Ninth IEEE International Conference on Computer Vision, 2003. pp. 734741 vol.2.

[7] A. Graves et al., "Speech recognition with deep recurrent neural networks", IEEE International Conference on Acoustics, Speech and Signal Processing, 2013.

[8] A. Graves et al., "Offline Handwriting Recognition with Multidimensional Recurrent Neural Networks", NIPS 2008

[9] Sundermeyer et al., "LSTM Neural Networks for Language Modeling.", Interspeech, 2012.

[10] S. Bell et al., "Inside-Outside Net: Detecting Objects in Context with Skip Pooling and Recurrent Neural Networks", CVPR 2016.

[11] Z. Deng et al., "Structure Inference Machines: Recurrent Neural Networks for Analyzing Relations in Group Activity Recognition", CVPR 2016.

[12] A. Alehi et al., "Social LSTM- Human Trajectory Prediction in Crowded Spaces", CVPR 2016.

[13] S. Ma et al., "Learning Activity Progression in LSTMs for Activity Detection and Early Detection", CVPR 2016.

[14] R. Stewart et al., "End-to-End People Detection in Crowded Scenes", CVPR 2016.

[15] Y. Bengio et al., "Learning Long-Term Dependencies with Gradient Descent is difficult", IEEE Transactions on Neural Networks, 1994.

[16] Hochreiter et al., "Long short-term memory.", Neural computation 9.8 (1997), 1735-1780.

[17] P. Pan et al., "Hierarchical Recurrent Neural Encoder for Video Representation with Application to Captioning", CVPR 2015.

[18] B. Singh et al., "A Multi-Stream Bi-Directional Recurrent Neural Network for Fine-Grained Action Detection", CVPR 2016.

[19] HC. Shinet al., "Learning to Read Chest X-Rays: Recurrent Neural Cascade Model for Automated Image Annotation", CVPR 2016.

[20] LeCun, Yann, et al. "Learning algorithms for classification: A comparison on handwritten digit recognition.", Neural networks: the statistical mechanics perspective 261 (1995): 276.

[21] Tom, et al. "Deep convolutional neural networks for pedestrian detection." Signal Processing: Image Communication (2016).

[22] Dalal et al., "Histograms of oriented gradients for human detection", CVPR 2005.

[23] Dollar et al., "Integral Channel Features", British Machine Vision Conference 2 (2009)

[24] Felzenszwalb et al., "Object detection with discriminatively trained part-based models", IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (2010) 16271645.

[25] Krizhevsky et al., "ImageNet Classification with Deep Convolutional Neural Networks", Neural Information Processing Systems (2012).

[26] P. Sermanetet al., "Pedestrian detection with unsupervised multi-stage feature learning.", In CVPR, 2013.

[27] W. Ouyang et al., "Joint deep learning for pedestrian detection.", In ICCV, 2013

[28] X. Zeng et al., "Multi-stage contextual deep learning for pedestrian detection", In ICCV, 2013.

[29] C. Szegedy et al., "Going deeper with convolutions", In CVPR 2015.

[30] K. Simonyan et al., "Very deep convolutional networks for large-scale image recognition.", In arXiv, 2014.

[31] R.Girshick et al., "Rich feature hierarchies for accurate object detection and semantic segmentation.", In CVPR, 2014.

[32] M. Liang et al., "Recurrent Convolutional Neural Network for Object Recognition", In CVPR 2015.

[33] Quoc et al., "A simple way to initialize recurrent networks of rectified linear units.", arXiv preprint arXiv:1504.00941 (2015).

[34] Dollar et al.,"Pedestrian detection: A benchmark.",CVPR 2009.

BIOGRAPHICAL STATEMENT

Vivek Arvind Balaji joined the University of Texas at Arlington in Fall 2014. Before joining UT Arlington, he received his B.Tech in Information Technology from Anna University, Chennai, Tamil Nadu, India in 2014. In UTA, he majored in Artificial Intelligence during his Masters Degree and he worked as a Machine Learning Intern in the Highly Autonomous Driving team at BMW of North America LLC. His current research interests include Machine Learning, Deep Learning, Computer Vision, Image Processing and Data Mining.