EMOTION RECOGNITION FOR DIGITAL SIGNAGE SYSTEMS

A Project Report

submitted by

CHILUMURU AKHIL REDDY

in partial fulfilment of requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS

MAY 2018

THESIS CERTIFICATE

This is to certify that the thesis titled **EMOTION RECOGNITION FOR DIGITAL SIGNAGE SYSTEMS**, submitted by **CHILUMURU AKHIL REDDY**, to the Indian Institute of Technology, Madras, for the award of the degree of **BACHELOR OF TECHNOLOGY**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof Kaushik Mitra
Project Guide
Assistant Professor
Dept. of Electrical Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 9th May 2018

ACKNOWLEDGEMENTS

This work would not have been possible without the guidance and the help of several people. I take this opportunity to extend my sincere gratitude to all those who made this thesis possible.

First, I would like to thank all my teachers who bestowed me with good academic knowledge. I am indebted to my advisor **Prof. Kaushik Mitra** whose expertise, generous guidance and support made it possible for me to work on a topic that was of great interest to me. I would also like to thank my lab mate and dear friend Raunak Kalani for sharing his valuable ideas and helping me whenever I am stuck with some problem.

I would like to thank my family for giving support and guidance all through my life. I would also like to thank all my friends and well-wishers for helping me in difficult times and being a good source of support and guidance.

ABSTRACT

The main objective of this report is the design and integration of various modules that help in betterment of advertising through Digital signage systems. This report describes Face Detection, tracking and design of emotion recognition. We present a method for classifying emotions from sequence of images. Our approach leverages on the recent success of Convolutional Neural Networks (CNN) and Long-short term memory (LSTM) Networks on Activity recognition.

Contents

List of Figures

Chapter 1

Introduction and Motivation

Modern applications of digital signage are interfaces to public or internal information, advertising, brand building and making enhanced customer experience. Digital signage displays have the advantage over static signs because they can display the multimedia content such as images, animations, video and audio. The content can be adapted in real time to a different context and audience, making it attractive for use at airports, hotels, universities, retail stores and various outdoor public spaces. However, the displayed content is frequently generic and uninteresting for observers. To make digital signage more effective as an information interface, the displayed content should be informative, dynamic and attractive. The actual attention that people pay to public displays is one of the key parameters of digital signage. Paying attention to public displays is a complex process, which depends on several criteria such as positioning of the display, display size, content format and content dynamics and most importantly, measuring the target's interpretation. Therefore, to maximize the attention to digital signage, these parameters should be considered already during the design phase of the digital signage system.

Facial emotions are important factors in human communication that help us understand the intentions of others. In general, people infer the emotional states of other people, such as joy, sadness, and anger using facial expressions and vocal tone. According to many surveys verbal components convey one-third of human communication, and non-verbal components convey two-thirds. Among several nonverbal components, by carrying emotional meaning, facial expressions are one of the main information channels in interpersonal communication.

The remainder of this report is organized as follows. In Chapter 2, the steps required to carry through Emotion recognition are described along with the previous works. Chapter 3 focuses on realizing emotion recognition as activity recognition and describes the architecture to achieve it. Chapter 4 proposes our method and the experiments carried out and their results.

Chapter 2

Previous Work

2.1 Steps in Emotion Recognition

For any type of recognition task, the first critical step is face tracking. This is followed by extraction of useful features after suitable preprocessing of raw data. Face tracking itself has to be done only after face detection. We will now review some of the existing techniques for face detection.

2.2 Face Detection

Face detection is a crucial step in various problems involving verification, identification, tracking, expression analysis etc. We are going to look at two deep learning based face detection algorithms *YOLO*(1) and *SSH*(2).

2.2.1 You Only Look Once: Unified, Real-Time Object Detection

YOLO is a generic object detection algorithm. For our purpose we have trained it on face data. The object detection is framed as single regression problem, straight from image pixels to bounding box coordinates and class probabilities. Hence the name YOLO i.e., you look (process) only once at an image to predict where the objects are.

Figure 2.1 shows an overview of the YOLO network architecture. YOLO has 24 convolutional layers followed by 2 fully connected layers.

Unified Detection

The network uses features from the entire image to predict each bounding box. It also predicts all bounding boxes across all classes for an image simultaneously. This means

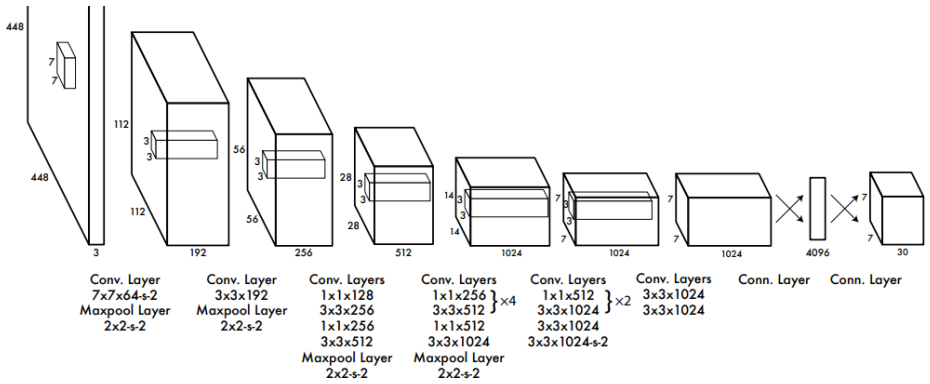


Figure 2.1: YOLO network architecture

it reasons globally about the full image and all the objects in the image. The YOLO design enables end-to-end training and realtime speeds while maintaining high average precision.

The system divides the input image into an S x S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts B bounding boxes and confidence scores for those boxes. These confidence scores reflect how confident the model is that the box contains an object and also how accurate it thinks the box is that it predicts. If no object exists in that cell, the confidence scores should be zero. Otherwise we want the confidence score to equal the intersection over union (IOU) between the predicted box and the ground truth. Each bounding box consists of 5 predictions: x, y, w, h, and confidence. The (x, y) coordinates represent the center of the box relative to the bounds of the grid cell. The width and height are predicted relative to the whole image. Finally the confidence prediction represents the IOU between the predicted box and any ground truth box.

Each grid cell also predicts C conditional class probabilities. These probabilities are conditioned on the grid cell containing an object. It only predicts one set of class probabilities per grid cell, regardless of the number of boxes B. At test time it multiplies the conditional class probabilities which gives us class-specific confidence scores for each box. These scores encode both the probability of that class appearing in the box and how well the predicted box fits the object and the individual box confidence predictions.

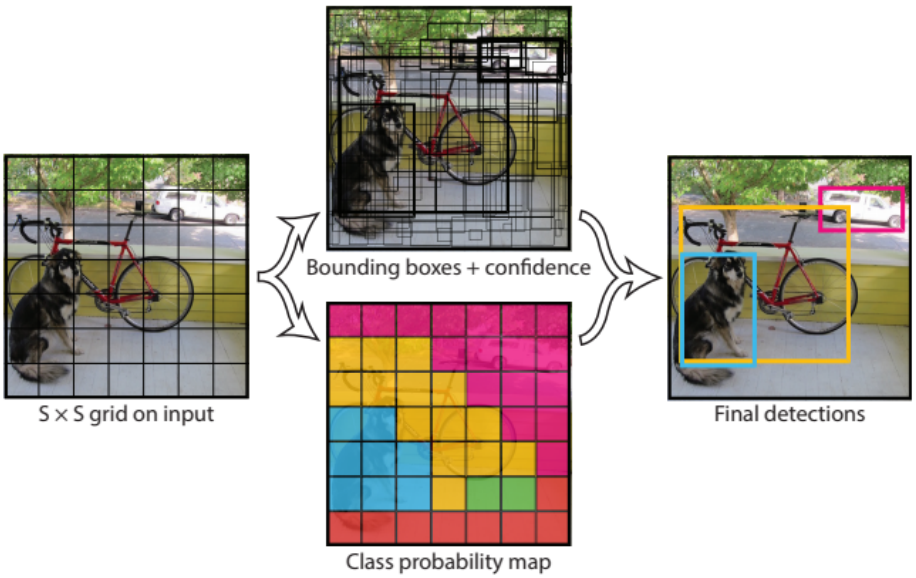


Figure 2.2: YOLO models detection as a regression problem. It divides the image into S x S grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as S . S . (B . 5 + C) tensor.

2.2.2 SSH: Single Stage Headless Face Detector

SSH performs detection in a single stage. Unlike the state-of-art CNN based detectors, SSH does not have a 'head'. The head is referred to the fully connected layers in VGG-16. It is able to achieve state-of-art performance without having the head of its underlying VGG-16 network. SSH is also scale-invariant by design. SSH detects faces from various depths of the underlying network. This is achieved by placing an efficient convolutional detection module on top of the layers with different strides, each of which is trained for an appropriate range of face scales.

Network Architecture

Figure 2.3, shows the general architecture of SSH. It is a fully convolutional network which localizes and classifies faces early on by adding a detection module on top of feature maps with strides of 8, 16, and 32, depicted as M1, M2 and M3 respectively to detect small, medium and large faces. The detection module consists of a convolutional binary classifier and a regressor for detecting faces and localizing them respectively.

To solve the localization sub-problem, SSH regresses a set of predefined bounding

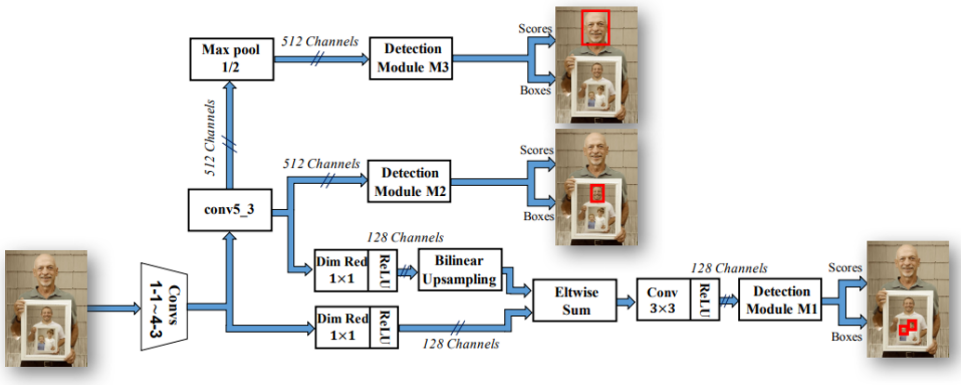
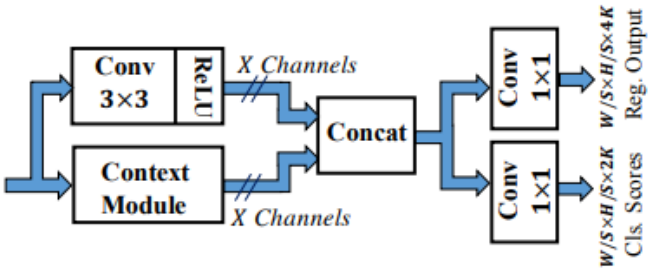
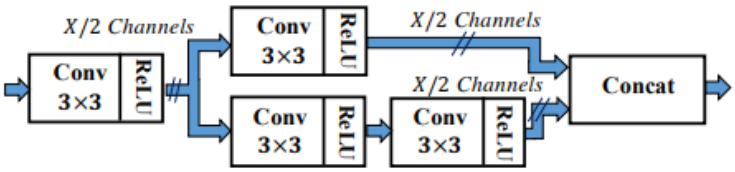


Figure 2.3: SSH network architecture



(a)



(b)

Figure 2.4: Detection and Context Modules

boxes called anchors, to the ground-truth faces. The anchors are defined in a dense sliding window fashion. At each sliding window location, K anchors are defined which have the same center as that window and different scales. For the detection module, a set of convolutional layers are deployed to extract features for face detection and localization as shown in Figure 2.4. This includes a simple context module to increase the effective receptive field. Finally, two convolutional layers perform bounding box regression and classification. At each convolution location , the classifier decides whether the windows at the filter center and corresponding to each of the scales contains a face. At each location during the convolution, the regressor predicts the required change in scale and translation to match each of the positive anchors to faces.

In unconstrained settings, faces in images have varying scales. SSH detects large

and small faces simultaneously in a single forward pass of the network. This gives it scale invariant design.

In state-of-art two stage detectors, it is common to incorporate context by enlarging the window around the candidate proposals. SSH mimics this strategy by applying a larger filter, which resembles increasing window size around proposals in two-staged detector.

2.2.3 Dataset and Comparison

SSH and YOLO are trained and tested on WIDER dataset. This dataset contains 32, 203 images with 393, 703 annotated faces, 158, 989 of which are in the train set, 39, 496 in the validation set and the rest are in the test set. This is one of the most challenging public face datasets mainly due to the wide variety of face scales and occlusion. We evaluate both SSH and YOLO on the test set. Figure 2.5, shows the test results visually.

(a) (b)

(c) (d)

Figure 2.5: YOLO vs SSH

As we can see, SSH is more versatile i.e., scale-invariant and works better. Hence, we choose SSH for the task of face detection.

2.3 Face Tracking

2.3.1 GOTURN: Generic Object Tracking using Regression networks

GOTURN(3) is an off-line tracker, that benefits from the large number of videos that are already available for off-line training. The tracker learns a generic relationship between object motion and appearance and can be used to track novel objects that do not appear in the training set. GOTURN tracks generic objects at 100fps. While most trackers take classification based approach using sliding windows, GOTURN uses a regression based approach requiring just a single feed forward pass through the network to regress directly the location of the target object.

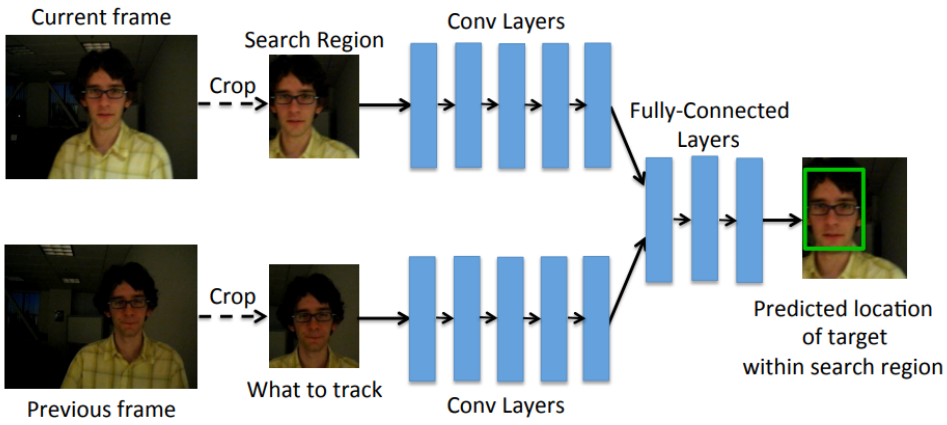


Figure 2.6: GOTURN network

The network only passes two images through the network, and the network regresses directly to the bounding box location of the target object.At a high level, we feed frames of a video into a neural network, and the network successively outputs the location of the tracked object in each frame.

What to track

We input an image of the target object into the network. We crop and scale the previous frame to be centered on the target object, as shown in Figure 2.6. This input allows our network to track novel objects that it has not seen before; the network will track whatever object is being input in this crop. We pad this crop to allow the network to

receive some contextual information about the surroundings of the target object. In more detail, suppose that in frame t-1, our tracker previously predicted that the target was located in a bounding box centered at c = (cx, cy) with a width of w and a height of h. At time t, we take a crop of frame t-1 centered at (cx, cy) with a width and height of k1.w and k1.h, respectively. This crop tells the network which object is being tracked. The value of k1 determines how much context the network will receive about the target object from the previous frame.

Where to look

To find the target object in the current frame, the tracker should know where the object was previously located. Since objects tend to move smoothly through space, the previous location of the object will provide a good guess of where the network should expect to currently find the object. We achieve this by choosing a search region in our current frame based on the object previous location. We crop the current frame using the search region and input this crop into our network, as shown in Figure 2.6. The goal of the network is then to regress to the location of the target object within the search region. In more detail, the crop of the current frame t is centered at c0 = (c0x, c0y),where c0 is the expected mean location of the target object. We set c0 = c, which is equivalent to a constant position motion model, although more sophisticated motion models can be used as well. The crop of the current frame has a width and height of k2 w and k2 h, respectively, where w and h are the width and height of the predicted bounding box in the previous frame, and k2 defines our search radius for the target object. In practice, we use k1 = k2 = 2. As long as the target object does not become occluded and is not moving too quickly, the target will be located within this region.

Network

For single-target tracking, an image-comparison tracking architecture is used, as shown in Figure 2.6. In this model, we input the target object as well as the search region each into a sequence of convolutional layers. The output of these convolutional layers is a set of features that capture a high-level representation of the image. The outputs of these convolutional layers are then fed through a number of fully connected layers. The role

of the fully connected layers is to compare the features from the target object to the features in the current frame to find where the target object has moved.

Between these frames, the object may have undergone a translation, rotation, lighting change, occlusion, or deformation. The function learned by the fully connected layers is thus a complex feature comparison which is learned through many examples to be robust to these various factors while outputting the relative motion of the tracked object. The convolutional layers in GOTURN are taken from the first five convolutional layers of the CaffeNet architecture.

Training

The network is trained with a combination of videos and still images. Videos contain a set of labeled frames. For each successive pair of frames in the training set, we crop the frames and we feed this pair of frames into the network and attempt to predict how the object has moved from the first frame to the second frame.

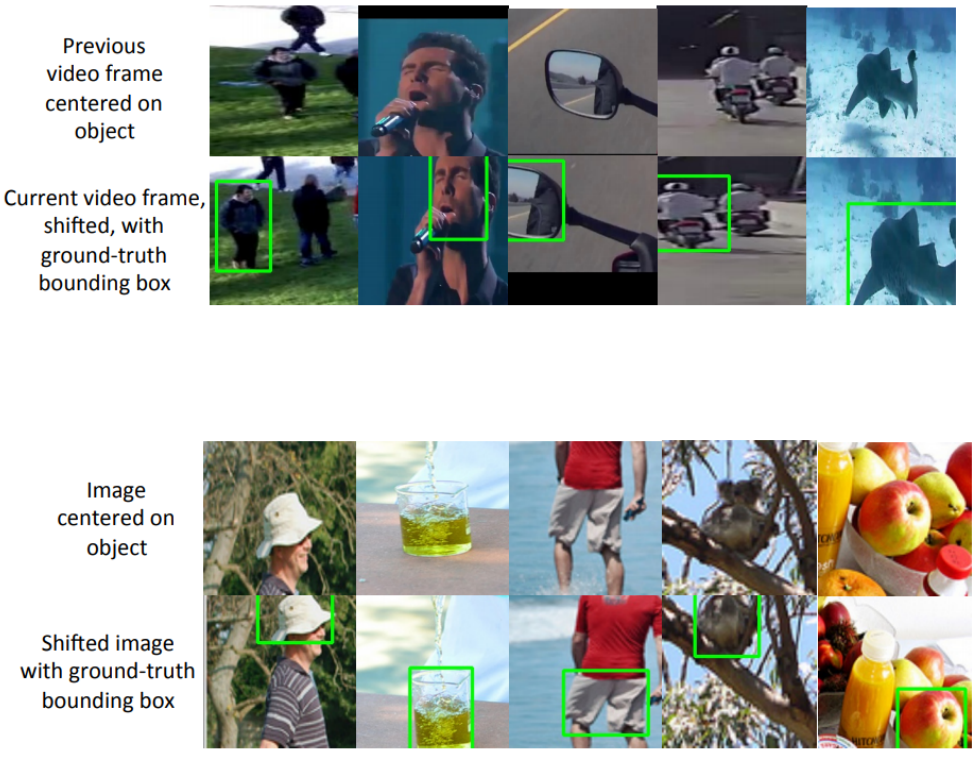


Figure 2.7: Training on videos and still images

For training on still images, we shift the image with small motions and attempt to predict how the object has moved in the new shifted image. This way the network takes

advantage of labeled still images in order to prevent overfitting.

Tracking

During test time, we initialize the tracker with a ground-truth bounding box from the first frame, as is standard practice for single-target tracking. At each subsequent frame t, we input crops from frame t-1 and frame t into the network to predict where the object is located in frame t. We continue to re-crop and feed pairs of frames into our network for the remainder of the video, and our network will track the movement of the target object throughout the entire video sequence.

2.4 Emotion Recognition

In general, humans express their emotions using facial expressions and vocal tone. Facial expressions are solely movements in facial muscles. In vision terms, it is a sequence of movements. In this chapter, we are going to focus on the problem of recognizing facial expressions.

2.4.1 Conventional Methods

In conventional Facial expression recognition, the recognition task is composed of three major steps, as shown in Figure 2.8:

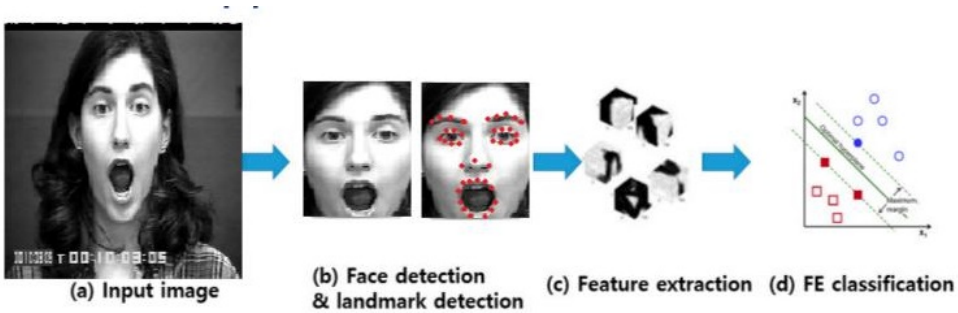


Figure 2.8: Figure shows conventional approach for Emotion Classification.

Face and Facial component detection, feature extraction and expression classification. First, a face image is detected from an input image, and facial landmarks are detected

from face region. Second, various spatial features are extracted from the facial components. Third, the pre-trained classifiers, produce the recognition result using the extracted features.

2.4.2 Deep learning methods

In contrast to conventional approaches using handcrafted features, deep learning approach highly reduces the dependence on face and condition specific models, and other pre-processing techniques by enabling end-to-end learning directly from the input images. Face emotion recognition can also be divided into two types based on whether it uses frame or video images.

2.4.3 Spatial information

Frame based facial recognition relies solely on static facial features obtained by extracting features from peak expression frames of image sequences. Convolutional Neural Networks (CNN) are used for this purpose. Figure 2.9 shows the procedure used by CNN-based facial emotion recognition.

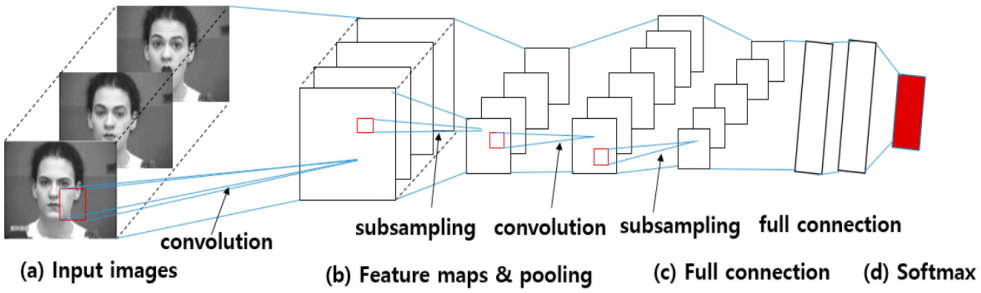


Figure 2.9: CNN based network that extracts only spatial information

The input image is convolved through a filter collection in the convolution layers to produce a feature map. Each feature map is then combined to fully connected networks, and the face expression is recognized as belonging to a particular class based on the output of the softmax algorithm.

2.4.4 Temporal Information

CNN based methods utilize only the static (spatial) information and cannot account for the temporal variations in a facial expression. To incorporate temporal features, we introduce Long sort-term memory (LSTM) a special type of deep leaning network and combine them with existing CNN network.

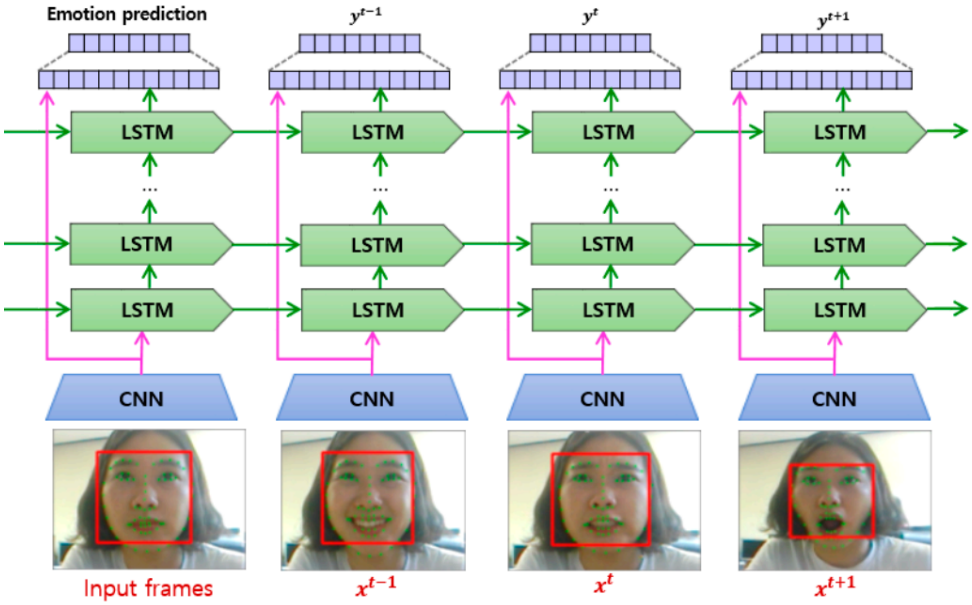


Figure 2.10: A general Hybrid CNN-LSTM structure

2.5 Activity recognition

(4) work proposes a Long-term Recurrent Convolutional Network (LRCN) model combining deep hierarchical visual feature extractor (CNN) with an LSTM model that can learn to recognize activities. (4) considers activity recognition as a sequential learning task with the goal of predicting a static output (label) from a sequential input (video). Figure 3.4, shows an overview of the network proposed by (4). We build our network inspired by (4). The general framework of the hybrid CNN-LSTM based approach will have a structure as shown in Figure 2.10. The basic framework of CNN-LSTM is to combine an LSTM with a deep hierarchical visual feature extractor such as a CNN model. Therefore, this hybrid model can learn to recognize and synthesize temporal dynamics for tasks involving sequential images. As shown in Figure 2.11, each visual

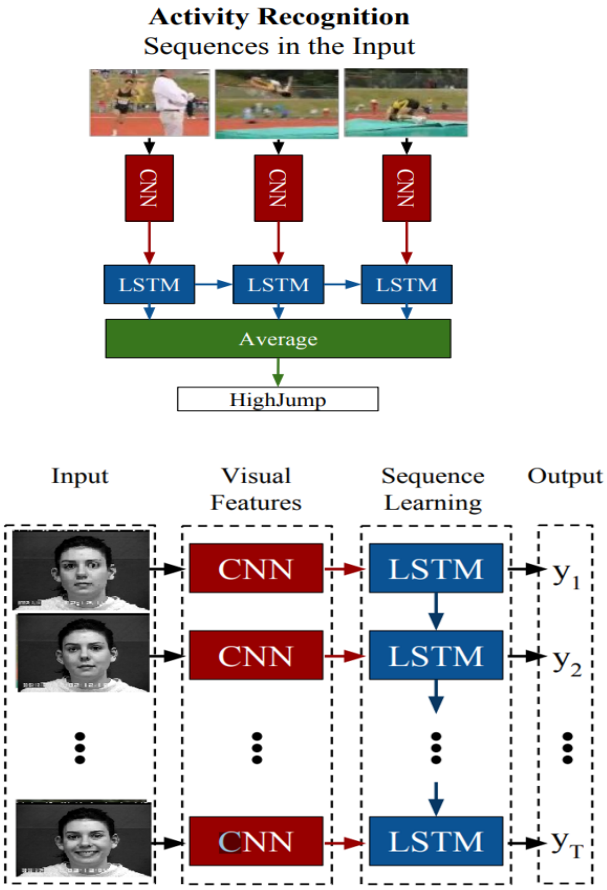


Figure 2.11: Emotion Recognition as Activity Recognition

feature is determined through a CNN is passed to the corresponding LSTM, and produces a fixed length vector representation. The outputs are then computed by applying softmax algorithm.

Chapter 3

Emotion Recognition as activity recognition

As mentioned before, emotion is basically an activity involving facial muscles. We base this as a central idea in building a suitable network for emotion recognition.

3.1 Dataset

The Extended Cohn-Kanade Dataset (CK+)(5) CK+ contains 593 video sequences of facial expressions, out of which 323 are labeled. The videos are collected from 210 adults in which 69% are female, 81% are Euro-American, 13% are Afro-American and 6% are from other groups. Participants are 18 to 50 years of age. A total of 7 basic expressions are labeled in the dataset: Anger, Contempt, Disgust, Fear, Happy, Sad, Surprise. All of the image sequences in CK+ start from the neutral face and end at the peak frame. We chose CK+ dataset for our experiments because it is one the only few Facial Expression Datasets that are publicly available. The dataset is pre-processed to have only the face-cropped videos.

3.2 Implementing Emotion Recognition

In this chapter, we propose a network that focuses to recognize facial expressions. Figure 3.2 shows an overview of the network.

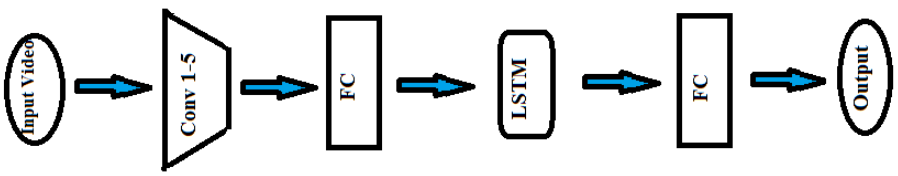


Figure 3.1: Emotion Recognition Network

Each frame in a sequence is the input to a single convolutional network(i.e;, the convnet weights are tied across time). We consider normalized RGB images as inputs to our

system. During training, videos are resized to 240 X 320 and we augment our data by using 227 X 227 crops and mirroring. Additionally, we train the Long-term Recurrent Convolutional Networks (LRCN) network with video clips of 8 frames, even though the CK+ dataset videos are generally much longer. For training with shorter video clips, we repeat the first frame(neutral frame) and augment it to make them at least of length 8. LRCN is trained to predict the video's class at each time step. To produce a single label prediction for an entire video clip, we average the label probabilities - the outputs of the network's softmax layer - across all frames and choose the most probable label. At test time, we extract 8 frame clips with a stride of 4 frames from each video and average across all clips from a single video. The CNN base of LRCN in our emotion recognition experiments is the *CaffeNet*(6) reference model (a minor variant of AlexNet).This gives a strong initialization to facilitate faster training and avoid overfitting to our small CK+ dataset.

Chapter 4

Experiments

The following experiments have been performed while training the network.

1. Training all layers: The base CNN model of LRCN is randomly initialized and all the layers including CNN are trained from scratch.
2. Pre-trained weights from CaffeNet: The base CNN model of LRCN is loaded with weights from CaffeNet and all the convolutional layers are locked from training phase. Only the Fully connected layers and LSTM are trained.
3. Re-training weights from CaffeNet: The base CNN model is reconstructed (made another copy) and initialized with weights from CaffeNet. This model is trained on CK+ dataset and the resulting trained weights are copied to the base CNN model of LRCN, and all the conv layers are locked. Only the fully connected layers and LSTM are trained.

4.1 Results

In theory, as the training data is very small, training all the layers would not be effective.

But to reaffirm this, we compare the test results for all the above mentioned experiments.

The results are shown in Figure 4.1.

Training all layers:

	Anger	Contempt	Disgust	Fear	Happy	Sad	Surprise
Anger	37	2	2	0	8	0	48
Contempt	11	72	11	0	0	0	5
Disgust	6	0	13	0	3	0	75
Fear	16	0	0	4	4	0	75
Happy	1	0	2	0	24	0	71
Sad	11	0	18	0	7	3	59
Surprise	2	1	0	0	3	0	92

Pre-trained:

	Anger	Contempt	Disgust	Fear	Happy	Sad	Surprise
Anger	13	0	2	0	0	4	80
Contempt	0	83	0	0	0	0	16
Disgust	3	0	36	0	0	0	60
Fear	0	0	0	20	0	4	75
Happy	1	0	1	0	13	1	82
Sad	3	0	3	0	0	11	81
Surprise	0	0	0	0	0	0	100

Re-training:

	Anger	Contempt	Disgust	Fear	Happy	Sad	Surprise
Anger	28	11	15	0	0	0	44
Contempt	0	94	0	0	0	0	5
Disgust	3	0	75	0	0	0	20
Fear	8	8	0	16	4	0	62
Happy	0	0	2	0	60	0	36
Sad	7	3	3	0	0	37	48
Surprise	0	1	0	0	1	0	97

Figure 4.1: Confusion Matrix for experiments

We therefore conclude that, Retraining CaffeNet gave us the best results.

4.2 Comparison

We now compare the results of LRCN model with other works. As we can see from Figure 4.2, we are able to achieve comparable results with lesser training data.

Emotion Recognition in the wild, Hassner T et al., ICMI 2015

	Anger	Disgust	Fear	Happy	Neutral	Sad	Surprise
Anger	**54**	2	16	10	2	5	7
Disgust	66	**0**	16	16	0	0	0
Fear	22	0	**19**	9	16	25	6
Happy	3	8	3	**75**	0	6	3
Neutral	4	4	4	4	**62**	16	2
Sad	9	5	12	14	14	**37**	5
Surprise	11	2	19	0	2	14	**50**

Kanade T et al., CVPRW 2010

	Anger	Contempt	Disgust	Fear	Happy	Sad	Surprise
Anger	**35**	0	40	0	5	5	15
Contempt	15	**25**	3	6	0	15	34
Disgust	8	2	**68**	0	15	5	0
Fear	8	13	0	**21**	21	8	26
Happy	0	0	0	0	**98**	1	0
Sad	28	24	4	12	0	**4**	28
Surprise	0	0	0	0	0	0	**100**

Results

	Anger	Contempt	Disgust	Fear	Happy	Sad	Surprise
Anger	**28**	11	15	0	0	0	44
Contempt	0	**94**	0	0	0	0	5
Disgust	3	0	**75**	0	0	0	20
Fear	8	8	0	**16**	4	0	62
Happy	0	0	2	0	**60**	0	36
Sad	7	3	3	0	0	**37**	48
Surprise	0	1	0	0	1	0	**97**

Figure 4.2: Comparing our results to other works

Bibliography

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection* arXiv:1506.02640, 2015.

[2] Mahyar Najibi, Pouya Samangouei, Rama Chellappa, Larry Davis. *SSH: Single Stage Headless Face Detector* International Conference on Computer Vision (ICCV), 2017.

[3] David Held, Sebastian Thrun, Silvio Savarese. *Learning to Track at 100 FPS with Deep Regression Networks* ECCV, 2016.

[4] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, Trevor Darrell. *Long-term Recurrent Convolutional Networks for Visual Recognition and Description* arXiv:1411.4389v2 [cs.CV], November 2014.

[5] Patrick Lucey, Jefferey F. Cohn, Takeo Kanade. *The Extended Cohn-Kanade Dataset (CK+): A complete dataset for action unit and emotion-specified expression* IEEE Conf. Computer Vision and Pattern Recognition Workshops (CVPRW), june 2010.

[6] Y. Jia, E. Shelhammer, J. Donahue, S. Kaayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. *Caffe: Convolutional architecture for fast feature embedding* arXiv:1408.5093, 2014.