SYNTHETIC VIDEO GENERATION - SCENE RECOGNITION AND INTEGRATION

MTP Report submitted to

Indian Institute of Technology Mandi

for the award of the degree

of

B. Tech

by

AMIT KUMAR

under the guidance of

Dr. Dileep A. D.

Dr. Renu M. Rameshan

SCHOOL OF COMPUTING AND ELECTRICAL ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MANDI

JUNE 2017

Acknowledgments

I wish to express my sincere gratitude to my mentors Dr. Dileep A.D and Dr. Renu M. Rameshan for mentoring me in my Major Technical Project.

Amit Kumar

Abstract

Humans are very curious species. This curiosity has made them touch the Moon and possibly Mars (who knows ?), another such fantasy of humans is developing intelligent systems which can function like humans. Significant research in fields of computer vision and machine learning in the past few decades have played a momentous part in fulfilling the above aspiration. Motivated by the above dream, this ambitious project is another step for the same.

There have been several projects and considerable amount of work done in generating summary [1], annotations or subtitles (text) from different kinds of videos however the other way round is less scrutinized field, which this project intends to explore. Applicable to sitcoms or other series where all scenes pertain to a few known locations. Given a new script (textual description) and corpus of videos of old episodes, a new video in accordance to script description has to be generated. In crux for generating the new video, frames from the existing database of episodes are picked and clubbed together justifying the content of script. This project is a composition of several challenges put together such as identifying and recognizing the people involved, their emotions, and location of a particular scene (background detection), picking up a matching scene, ensuring continuity between successive frames, etc.

Contents

List of Figures

Chapter 1

Introduction

Generating a video which is not actually shot is not an easy task. Also making it conferring with the context going around in the scene makes it much more challenging This work uses concepts from Computer Vision and Machine Learning to solve the problem of synthetically generating a description suitable video. The whole work is divided into several independent modules like face recognition,scene recognition, activity recognition, emotion recognition etc. which when combined makes up whole working model. This report talks about scene recognition, emotion recognition and integration of all the modules. Character recognition, Activity recognition is elaborated in the report [4] titled 'Synthetic Video Recognition - Character and Activity Recognition'. Combining these two adds up to the completion of the whole work 'Synthetic Video Generation'. This project focuses on generating new episodes' video of a sitcom TV series (Friends here) using its previous episodes as a training dataset. A sitcom generally revolves around a fixed set of locations and people who remains almost same and finitely small. So the model can be trained to learn the locations and identify the characters without much human effort of building labeled training dataset. Also the places and people mentioned in the script will most likely be present in the videos provided as an input, from which frames can be searched and picked to generate the new video as per the script.

1.1 Problem Statement

To develop a model which can generate synthetic video, a video which is not originally shot but developed from picking frames from a corpus of videos already shot. The model will be provided with a new script which describes the video to be generated and a corpus of videos. The model has to come up with a new videos by picking up the frames from the corpus which best describes the scenario mentioned in the script.

The model should consider various parameters like locations of the shot, continuous frames disparity, action, orientation, emotion of characters while developing the video. The model should be intelligent enough to find out the people it should consider for the shot and pick those frames which contain those people in it.

The location of the picked frame has to match with the location described in the script. The location should not frequently change and should change only at the end of the shot boundary. The continuous frame disparity has to be minimized in order to give viewer a smooth and pleasant experience.The action, orientation and emotion of the people involved in the frame should match with the semantic of the scene and the subtitles.

The final generated video should be comfortable enough in viewing and must provide a pleasant and smooth viewing experience.

1.2 Motivation

Generating a caricature from given verbal description of face etc is a challenging task even for a human and requires the person to be artful to generate an appealing caricature, in a similar manner the given problem requires the system to be intelligent enough to grasp the details from the description and generate it's ćaricature˝ the output video.

Consolidating all the matching frames to form new video requires frame-by-frame analysis and segregation of the videos from the corpus given as input. While generating a video by consolidation of frames sounds straightforward, it is in no way a plain sailing task and requires an in depth work spitting into several subproblems, two of them currently being focused are character identification and location recognition. Each of them is being individually tackled and is reflected in overall system as cascaded models of one succeeding the

other. An approach consists of using Deep Neural Networks(DNN). In this approach, raw image representation of frames are used as input surrounded by preceding and succeeding frames. This also facilitates the further use of Convolutional Neural Networks(CNN). Incentive for using them is that they learn features rather than requiring hand-crafted features which would be advantageous in complex scenarios like these. Another motivation for the use of DNNs and CNNs is their ability to leverage massive amounts of training data and improve performance with so. This results in the creation of a highly scalable system.

1.3 Input Specifications

The sitcom chosen for this project is Friends [5] created by David Crane and Marta Kauffman. The series revolves around 6 central characters namely Rachel Green, Ross Geller, Monica Geller, Joey Tribbiani, Chandler Bing and Phoebe Buffay, who are friends, living off of one another in the heart of New York City. Over the course of ten years, this average group of buddies goes through massive mayhem, family trouble, past and future romances, fights, laughs, tears and surprises as they learn what it really means to be a friend. The major locations in the show are Central Perk, a Manhattan coffeehouse where the group often hangs out; when not there, the six are usually at Monica and Rachel's nearby West Village apartment, or Joey and Chandler's across the hall. The series extends for 10 season with each season having 24 episodes, each stretching around 23 minutes in general.

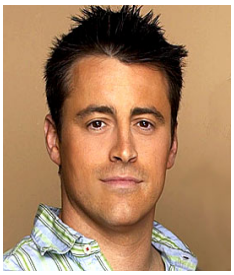


Fig. 1.1: Joey Tribbiani

Fig. 1.2: Chandler Bing

Fig. 1.3: Ross Geller

Fig. 1.4: Rachel Green

Fig. 1.5: Monica Geller

Fig. 1.6: Phoebe Buffay

Characters from Friends

Fig. 1.7: Central Perk

Fig. 1.8: Monica and Rachel's Apartment

Fig. 1.9: Chandler and Joey's Apartment

Major scene locations in Friends

Chapter 2

Background and Related Work

The videos on internet are growing day by day at a very huge rate processing these huge chunks of videos demands plenty of resources like time, man power, and hardware storage. Video summarization plays an important role in this context it summarizes the content of video and provide the summary and context of it. [6] But can a video which is not originally shot be constructed given the summary of the video in the text format, this is what we are dealing in this work. There is very less work done in this area due to its less application and the challenges it creates.

A recent development in the field of captioning a image locally is carried out in Dense Captioning. Dense Captioning task requires a computer vision system to both localize and describe salient regions in images in natural language. The dense captioning task generalizes object detection when the descriptions consist of a single word, and Image Captioning when one predicted region covers the full image. To perform the localization and description task jointly dense cap has proposed a Fully Convolutional Localization Network (FCLN) architecture which processes an image with a single, efficient forward pass, requires no external regions proposals, and can be trained end-to-end with a single round of optimization. The architecture is composed of a Convolutional Network, a novel dense localization layer, and Recurrent Neural Network language model that generates the label sequences. [7]

Convolutional neural networks have achieved great success for image classification. In the recent ILSVRC competition [8],several successful architectures were proposed for object recognition, such as GoogLeNet [9] and VGGNet [10]. These models have learnt the

features which are important for classification of different images. These architecture have many layers, each layer extract some feature stating from low level general features to very task specific features at the end. These features can be used in many tasks where images are involved. However, directly adapting these models trained on the ImageNet dataset to the task of scene recognition cannot yield good performance. Also training complicated VGGNets on a large-scale scene dataset is non-trivial, which requires large computational resource and numerous training skills.Instead one can use the low level features extracted by these model and train only few bottom layers to perform the task of other classification related to image. [11]

Emotion Recognition of people in the image is done after detecting the face, [12] finding features from the face and then classifying into different emotions based on the features. Deep Neural network are also available for emotion recognition but face detection and localization is required before passing the image to the network. [13] For further improving the accuracy of the emotion recognition model face registration is required. Face registration requires finding landmarks points from the face so that fix eye points of all the images can be fixed the same pixels by rotating,scaling and shifting the image. [14]

Chapter 3

Scene Recognition

Scene Recognition is one of the important module used in the work. Once the script/subtitles are provided and the model has identified the location of the shot to develop, then for making a video machine has to know the location of each and every frame belonging to training set.

There are broadly 4 classes made for the locations of Friends T.V. series, these are Monica's apartment, Chandler's and Joey's apartment, Central Perk cafe and one class which represents locations not belongs to any of the above. A convolutional Neural network is used to classify frames of training set in one of he 4 classes described above. Training a deep neural network from scratch requires a large amount of time and resources to converge. A neural network requires huge labeled training data for avoiding overfitting and we do not have labeled training data. Instead of building a huge labeled training data manually a more refined approach would be to leverage a network pre-trained on a large dataset. Such a network would have already learned features that are useful for most computer vision problems, and leveraging such features would allow us to reach a better accuracy than any method that would only rely on the small available data.

3.1 VGG16 Model Trained on ImageNet

We are using the VGG16 architecture, pre-trained on the ImageNet dataset.VGG is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image

Recognition" [2]. The model achieves 92.7 top-5 test accuracy in ImageNet.The architecture of the model is explained in the figure. 3.1

The ImageNet dataset [15] contains many classes related to indoor and outdoor scene among its total of 1000 classes, this model will already have learned features that are relevant to our classification problem. We are using keras library on theano for our neural net training and classification.

First of all a labeled training dataset having 300 images for each class is manually prepared by picking up frames and labeling after parsing the episodes of the friends. We are using the same weights of the VGG16 model upto fully-connected layer.The layers doing convolution part in the model are untouched and its weights are not modified. We run this model on our training and validation data once, recording the output which is the last activation maps before the fully-connected layers of the VGG16 model and stores it as further features for training and validation data. Then we will train a small fully-connected model on top of the stored features. [3] The model for this training is explained in figure. 4.5

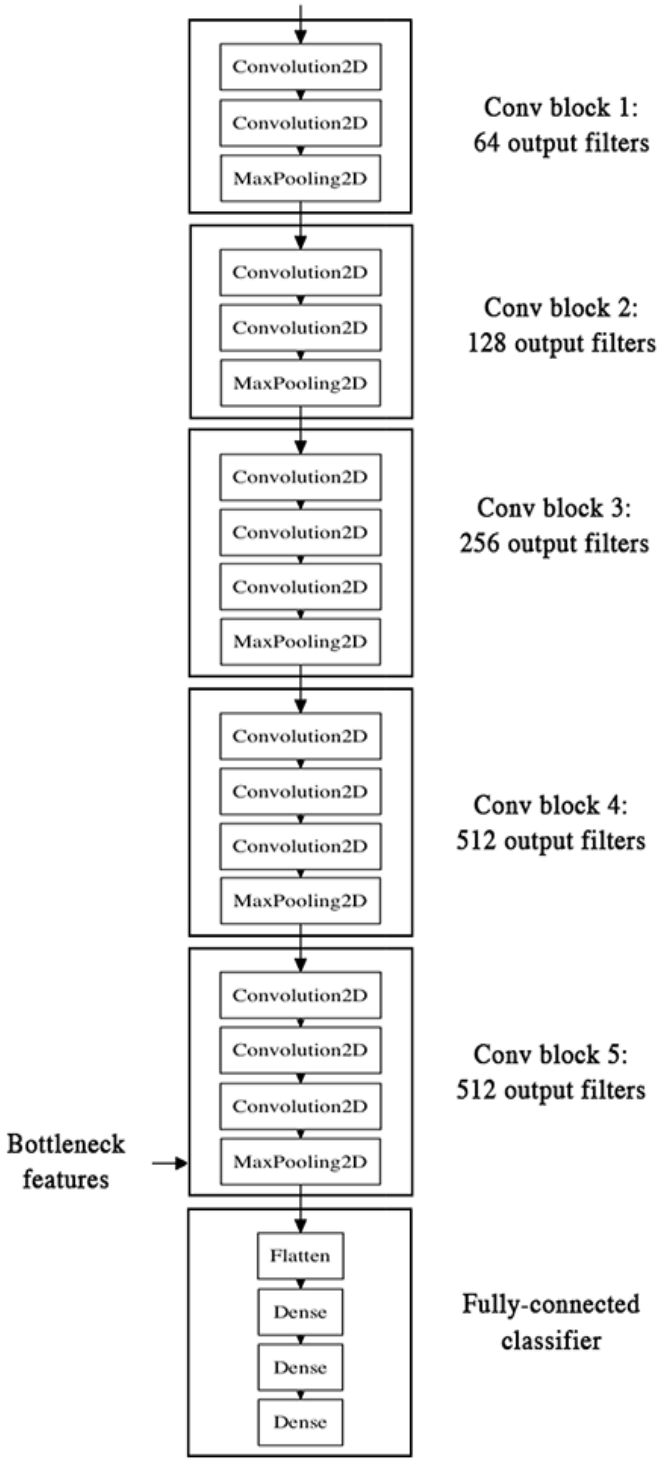


Fig. 3.1: VGG16 arcitecture [3]

To further improve our previous result, we tried to fine-tune the last convolutional block of the VGG16 model and the last fully connected layer. Fine-tuning consist in starting from a trained network, then re-training it on a new dataset using very small weight updates. In our case, this can be done in 3 steps:

- Instantiate the convolutional base of VGG16 and load its weights
- Add our previously defined fully-connected model on top, and load its weights
- Freeze the layers of the VGG16 model up to the last convolutional block

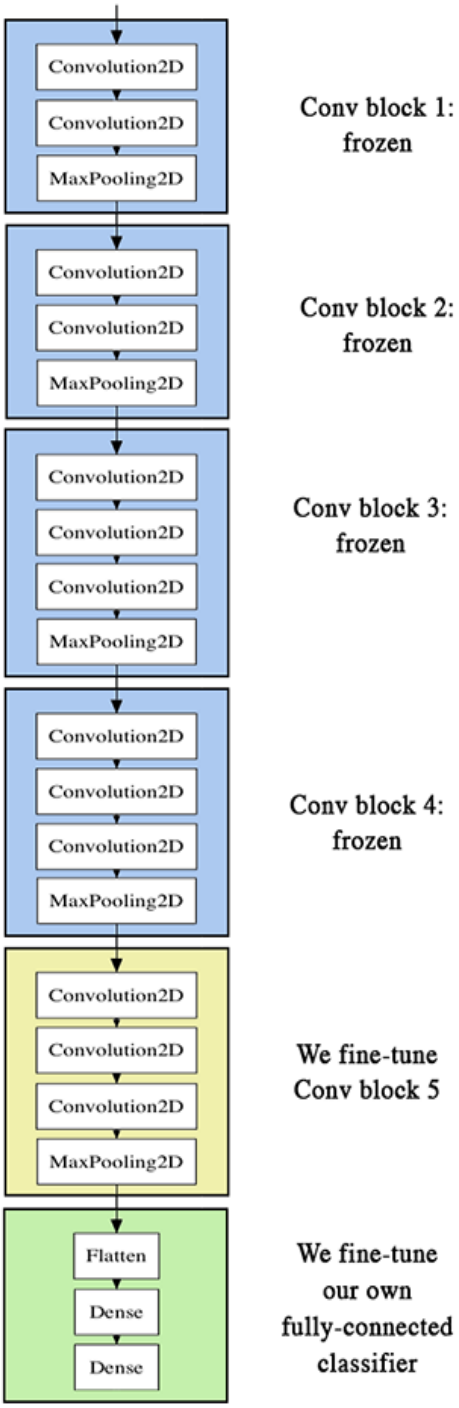


Fig. 3.2: Training convolutional layer of VGG16 architecture [3]

In order to perform fine-tuning, we initialised the weights of all layers with the trained weights. One should not randomly initialize weights of any layer in model rather use the weights of the pretrained network as initial weights, this is because the large gradient updates triggered by the randomly initialized weights would wreck the learned weights in the convolutional base and the model would perform bad. We can retrain from any layer, but we choose to only fine-tune the last convolutional block rather than the entire network in order to prevent overfitting.The entire network would have a very large entropic capacity and thus a strong tendency to overfit. The features learned by low-level convolutional blocks are more general, less abstract than those found higher-up, so it is sensible to keep the first few blocks fixed and only fine-tune the last one. This will fix more general features and train only the layer which give more specialized features.

Fine-tuning is done with a very slow learning rate, with the SGD optimizer keeping track of momentum and velocity of updation. This is to make sure that the magnitude of the updates stays very small, so as not to wreck the previously learned features.

Loss function - The loss function used in training the neural network is the cross entropy loss. The loss expression looks like

$$H(\mathbf{p}, \mathbf{q}) = \frac{-1}{N} \sum_{x=1}^{N} p(x) \log q(x) \tag{3.1}$$

where **p** is the true distribution and **q** is estimated distribution. **q** is the function which denotes the softmax classifier values.

$$q(y_i|x_i, \mathbf{W}) = \frac{e^{f_{y_i}}}{\sum_j e^{f_{y_i}}} \tag{3.2}$$

It can be interpreted as the (normalized) probability assigned to the correct label y_i given the image x_i and parameterized by **W**.

3.2 Classificationof images belongs to New Image class

The fouth class new image class has all the images which are not from any of the class Monica's aprartment, Chandler's apartment and Central Perk cafe. These class has a huge variety of images ranging from streets,other people home,offices etc. For classification of images to this class we explored two approaches they are -

- Building training data for the new image class by picking variety of images which are not from the above 3 classes and training the model for 4 classes.
- Not making seperate training data for new image class. Training model only to classify above three classes and do thresholding to classify images with low confidence score to new image class. We have used sigmoid function in out last layer and used a threshold of 0.5.

From out experimentation we found out that the approach 2 gives good result than approach 1 due to huge variety in new class images.

3.3 VGG Model Trained on PlacesNet [2]

We were using the model that was trained on ImageNet dataset, it has some classes of scene and places in it. But to get a better classification result we used VGG Model trained on PlacesNet.

MIT Computer Science and Artificial Intelligence Laboratory provides convolutional neural network models trained on scene-centric database called Places, with 205 scene categories and 2.5 millions of images with a category label [2]. PlacesNet is a caffemodel i.e the trained weights are in caffe. This VGG16 model has 5 convolutional blocks and 3 fully connected layer.

This paper [16] compares the performance of different models for classifying the Places205 data. They show that Places205-VGG model gives best performance with 0.746 accuracy followed by Places205-GoogleNet with 0.716 accuracy, followed by Places205-AlexNet with 0.675 accuracy, followed by ImageNet-AlexNet features+SVM with 0.496 accuracy. So we used PlacesNet VGG Model for improving the accuracy of our classification task.

We used caffe toolbox and PlacesNet VGG16 model to extract representation from training,validation and testing data class. The representation extracted are from first fully connected layer(i.e fc6 layer).Each image is now represented by 4096 dimension vector. We took the new representation any trained a fully connected neural network to classify images. The neural network has 4096 input nodes, the second layer has 256 node and output layer has 3 nodes(central perk cafe,monica apartment,chandler apartment). We used fixed thresholding to segregate and classify images which do not belong to any of the above three class.

Loss function - The loss function used is shown in equation 3.2.

Sigmoid Function - Further performance was improved by intoducing sigmoid function as an alternative for softmax function in the last output layer.

$$q(y_i|x_i, \mathbf{W}) = \frac{1}{1 + e^{-f_{y_i}}} \tag{3.3}$$

3.4 Results

For testing our classifier we took 600 images from an episode and classified it in the above described four classes. The result of the prediction is shown below 4.1. The rows of the matrix represents the actual class label and column represents the predicted class. Images belongs to new class represents the places which is not Central cafe,Chandler apartment or Monica's apartment. Based on the matrix below we calculate the accuracy of each class and is shown in the last column of the table. Some example of the classification is shown in images. 3.3 3.5 3.7 3.9

Table 3.1: Confusion Matrix and Accuracy

Actual	Predicted				-
-	Central Cafe	Chandler Apartment	Monica Apartment	New Place	Accuracy
Central Cafe	96	0	0	2	0.98
Chandler Apartment	3	44	0	10	0.77
Monica Apartment	8	4	201	57	0.74
New Place	17	47	37	74	0.42

Ground Truth: Central Cafe

Fig. 3.3: Classified

Fig. 3.4: Misclassified

Fig. 3.5: Classified

Fig. 3.6: Misclassified

Ground Truth: Chandler's Apartment

Fig. 3.7: Classified

Fig. 3.8: Misclassified

Ground Truth: Monica's Apartment

Fig. 3.9: Classified

Fig. 3.10: Misclassified

Ground Truth: New Class

The images which belongs to Central cafe is classified with good accuracy than the images belongs to Monica's apartment and Chandler's apartment. The Chandler's apartment class and monica's apartment class both are living room and have lot of common features. The central cafe class has more distictive feature as it is a cafe and other two classes are living rooms. So the model is unable to distinct chandler's apartment and monica's apartment

with high accuracy. The New image class has images which do not belongs to any of the above classes, these images include working place of Monica, Chandler's office, streets, Ross's apartment etc. Many images which bolongs to Ross's apartment is being classified as Chandler's apartment class as both of them are living room and have many similar features. Some of the images which belongs to Chandler's and Monica's apartment are classified in new image class because of their low confidence score in its actual class.

Chapter 4

Emotion Recognition

Emotion Recognition is module used in the work to tag each character in the frame with the emotion it is showing. Once the script/subtitles are provided, then for making a video the model has to know the emotion of each people so that it can make the video considering the emotion attached in the script. There are usually 7 classes of the emotion i.e. Angry,Surprised,Neutral etc.Several approaches has been tried to do the task of the emotion classification.HOG and LBP based feature descriptor followed by SVM classifier is used for classification of the emotions.A convolutional Neural network approach is also used to classify frames into appropriate emotions.

First of all faces are detected in the image using dlib,followed by face landmark extraction and face registration.Then several features are extracted from the image like HOG,LBP followed by SVM classification.

4.1 Face Detection

Following frame-by-frame approach, the very first step is detection of a face in a particular frame. For face detection we use adaptation of histogram of gradients [17]. We split up the image in grids of 16x16 pixels each. In each square, we count up how many gradients point in each major direction. Then we replace that square in the image with the arrow directions that were the strongest. This is how we get an histograms of gradient image. Now we find the detector pattern in our image. For this we use dlib's [18] implementation.

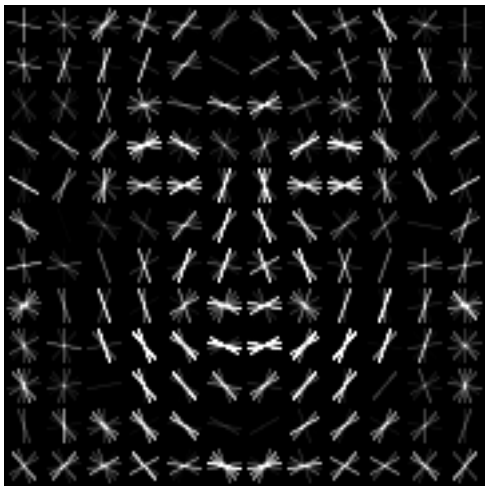


Fig. 4.1: Learned HOG detector which looks like a face.
Image Courtesy: [18]

4.2 Landmark Detection

The face detection portion returns a list of bounding boxes around the faces in an image that can be under different pose and illumination conditions. Humans can easily recognize faces that could be looking in different directions or under different illumination conditions, but computers would not know the shape of the face and its parts. To deal with this potential issue we will try to track the eyes and lips and the whole structure of the face, later we will find out the eye centre for face registration. To do this, we are using an algorithm called face landmark estimation [19] which is a simple 2D affine transformation to make the eyes and nose appear in similar locations. The basic idea is to use 68 specific points (called landmarks) that exist on every faceŁthe top of the chin, the outside edge of each eye, the inner edge of each eyebrow, etc.

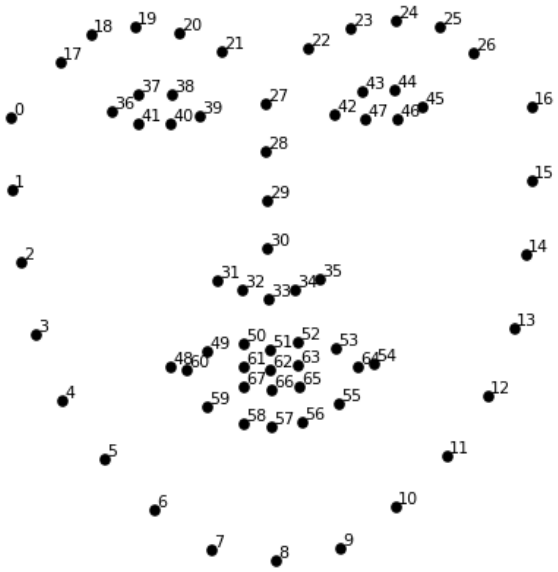


Fig. 4.2: The 68 landmarks located on each face.
Image Courtesy: [20]

4.3 Face Registration based on eye

Left and Right Eye centre are calculated taking average of the x and y coordinates of all the landmark points surrounds the eye region. Once the face is detected and the landmarks points are located face registration is performed. The faces can be in different orientation and there may be the variation in the scaling of the faces.The toolbox [14] provides approach to register the face based on the cente of the eyes. In the above approach the centre of the eyes of all the image is shifted at a fixed pixel for all the images. And the width and height of the face is shifted in the same way. In plane rotation of the face is also adjusted by rotating the face such that the eyes goes to the fixed place in the image.

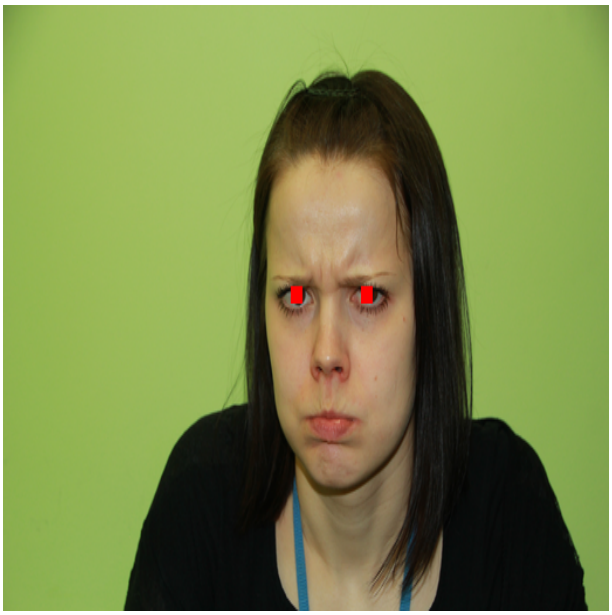


Fig. 4.3: Located eye centre using 64 landmarks points
Image Courtesy: [21]

4.4 Feature Extraction

4.4.1 LBP Feature

The Local Binary Patterns algorithm is wiely ised for 2D texture analysis. This algorithm compares each pixel with its neighborhood, finds out and summarizes the local structure in an image. Firstly, we convert the input color image to grayscale, as LBP works on grayscale images. We iterate on each and every pixel, for a pixel we consider it as center and threshold its neighbors against. If the intensity of the center pixel is greater-equal its neighbor, we label the neighbourhood pixel with 1 and 0 if not. We'll end up with a binary number for each pixel.Usually we take 8 neighbour pixel for a centre pixel. With 8 surrounding pixels we'll end up with 256 possible combinations, which are called Local Binary Patterns or sometimes abbreviated as LBP codes. We can start from any neighbouring pixel and then we can transverse either in clockwise or anti-clockwise direction but we must use the same order for all the pixels.Traversing in a particular direction with a fixed starting point chooses unique code among the 256 different possibilities. The first LBP operator uses a fixed 3 x 3 neighborhood,t his description enables you to capture very fine grained details in images. In

fact the authors were able to compete with state of the art results for texture classification. All the spatial information is lost if we throw all features into a single histogram. In tasks like face detection minute spatial information is very useful, so it has to be incorporated into the histogram somehow. The representation psroposed by Ahonen et al. in Face Recognition with Local Binary Patterns is to divide the LBP image into grids and build a histogram of each cell seperately. [22]

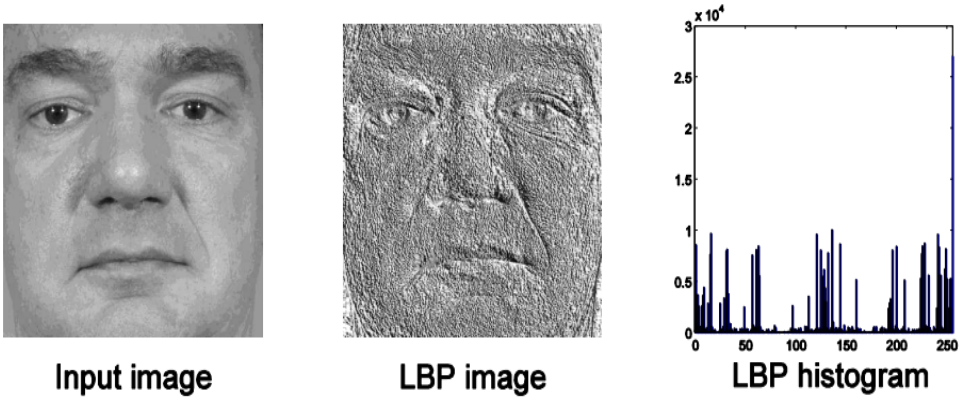


Fig. 4.4: LBP histogram feature
Image Courtesy: [22]

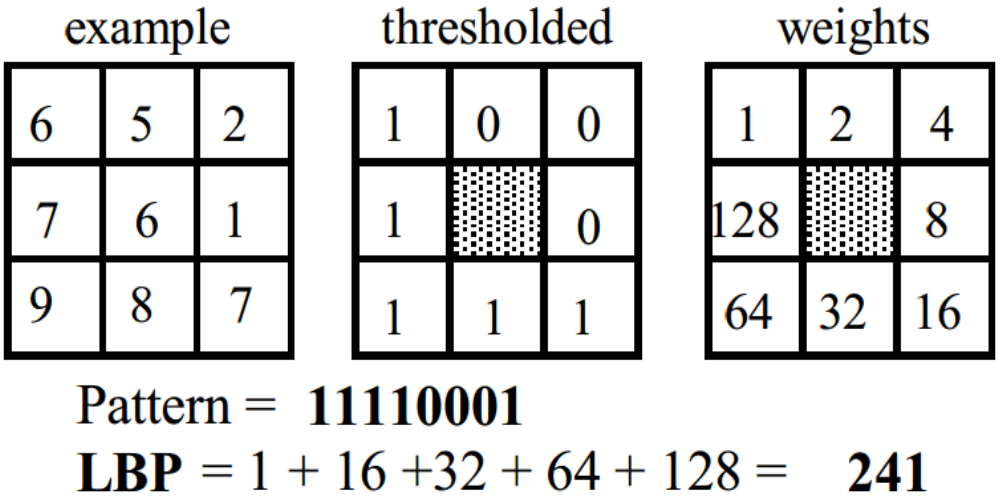


Fig. 4.5: LBP Code calculation
Image Courtesy: [22]

4.4.2 HOG Feature

The HOG features are widely use for object detection, it stands for Histograms of Oriented Gradients. It is a type of feature descriptor. It decomposes an image into small squared cells, computes the histogram of oriented gradients in each cell, normalizes the result using a block-wise pattern, and return a descriptor for each cell. This feature descriptor is to generalize the object in such a way that the same object produces as close as possible to the same feature descriptor when viewed under different conditions. This makes the classification task easier.

HOG uses a sliding detection window which is moved around the image. At each position of the detector window, a HOG descriptor is computed for the detection window. Generally HOG descriptor is computed on 8x8 patch. These patches can be organized into overlapping blocks or non-overlapping blocks . If overlapping block is used the feature vector will be of greater size compared to non-overlapping window. To compute HOG descriptor within a cell, we compute the gradient vector at each pixel. We take the 64 gradient vectors (in our 8x8 pixel cell) and generally put them into a 9-bin histogram. The Histogram ranges from 0 to 180 degrees, so there are 20 degrees per bin. For each gradient vector, its contribution to the histogram is given by the magnitude of the vector (so stronger gradients

have a bigger impact on the histogram). We split the contribution between the two closest bins. So, for example, if a gradient vector has an angle of 85 degrees, then we add 1/4th of its magnitude to the bin centered at 70 degrees, and 3/4ths of its magnitude to the bin centered at 90. [23]

Fig. 4.6: Image of a house
Image Courtesy: [23]

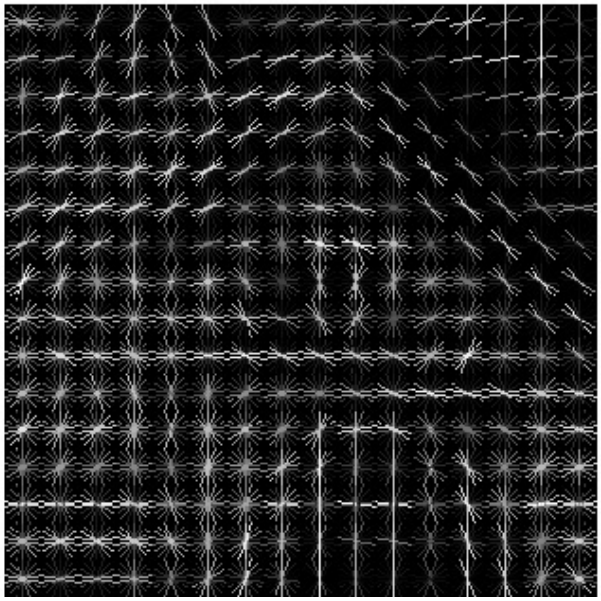


Fig. 4.7: HOG feature of a house

Image Courtesy: [23]

4.4.3 Deep CNN Feature

We used the VGG model that was fine-tuned on EmotiW dataset [13]. EmotiW database is a database that has images belongs to 6 major emotions i.e Anger, Disgust, Fear, Happiness, Sad, Surprise. The codalab emotion challenge database has 50 emotion classes. We followed the approach explained in section[3.1] for leveraging the model [13] and training a new model to classify images into the 50 required emotions required for the codalab challenge [21]. We took features from fc6 layer, 4046 dimentional features and trained a svm to classify the images.

4.5 Classification

This module of emotion recognition involve classifying image into its emotion class. Although any of the suitable classifier capable of classifying given feature vectors.In this case however we use Linear Support Vector machines (LinearSVM). Support vector machines (SVM) are a group of supervised learning methods that can be applied to classification or regression. The idea behind SVMs is to make use of a (nonlinear) mapping function Φ that transforms data in input space to data in feature space in such a way as to render a problem linearly separable. The SVM then automatically discovers the optimal separating hyperplane. The function Φ is realized by using a Mercer kernel. This is also known as the kernel trick [24].

These provide us advantages such as they are effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples. (which is the case here while training on small sample sizes). The setback lies in generating class wise probabilities or confidence score for a sample for each class, this involves slightly more computation than other classifiers which directly work upon these scores. For the implementation of above we use scikit-learn [25] a free software machine learning library for the Python programming language.

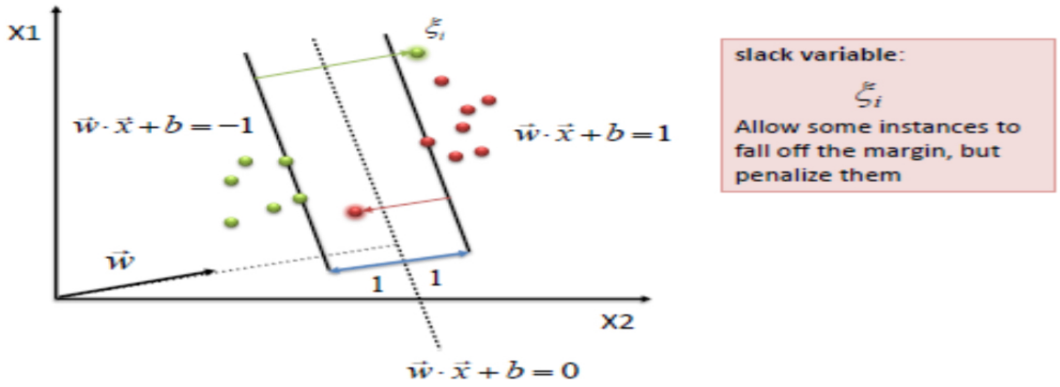


Fig. 4.8: Constructing the optimal separating hyperplane for linearly separable data.

4.6 Result

We build model for codalab emotion recognition database. [21] We have to automatic recognise multi-emotions from facial expressions and handle analysis of micro emotions.The database includes 31250 facial faces with different emotions of 125 subjects whose gender distribution is almost uniform. Each subject acts 50 different emotions and for each of these emotions 5 samples have been taken by Canon 60D camera under uniform lightening condition with relatively unchanged background and resolution of 5184x3456 per sample.The number of emotion classes are 50 in the form of complementary emotion and Dominant emotion.The 7 major emotions were angry, conempt, disgust, fear, happy, sad, surprise. All these 7 emotion permute to form 49 emotions of complementary,dominant type and the 50th emotion was the neutral emotion.

Table 4.1: Performance on Codalab emotion database

Classification Type	Misclassification
HOG + SVM	92.37
LBP + SVM	91.89
CNN fc6 + SVM	90.86
(HOG+LBP) + SVM	86.22

Fig. 4.9: Example of classes in emotion challenge.
Image Courtesy: [21]

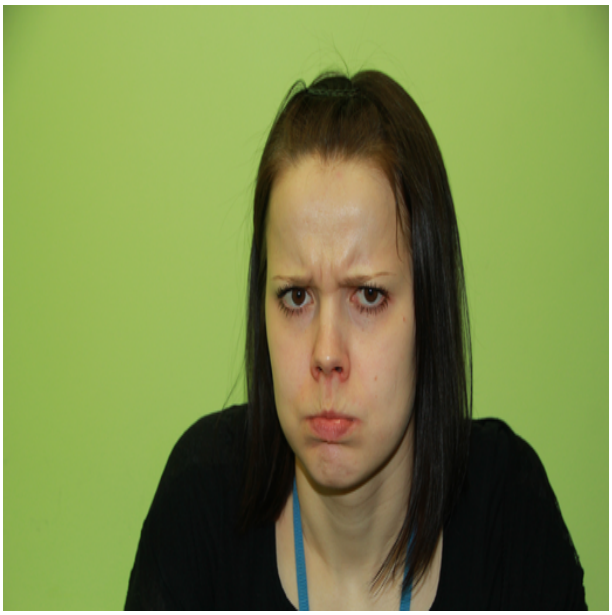


Fig. 4.10: An image to be classifed
Image Courtesy: [21]

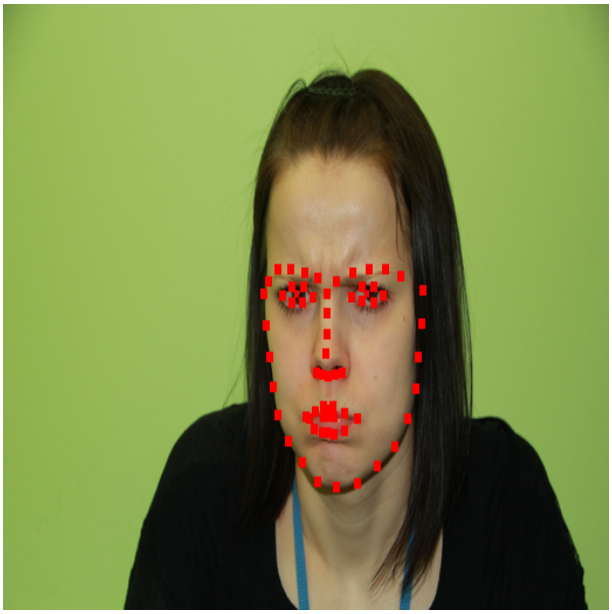


Fig. 4.11: Landmarks Detection from the facial region
Image Courtesy: [21]

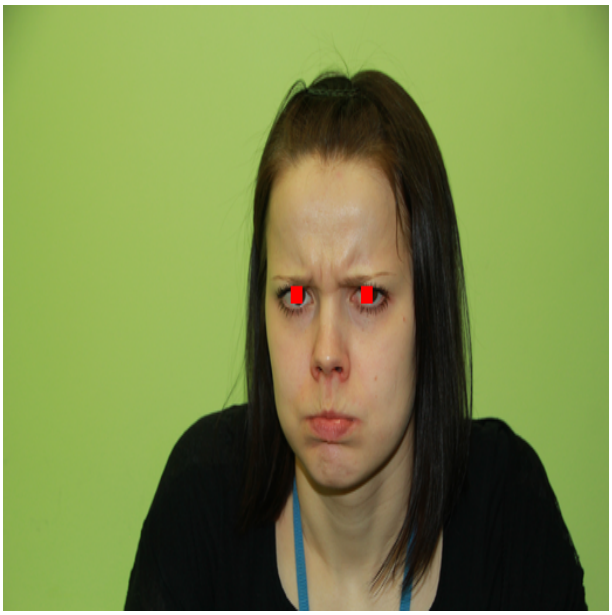


Fig. 4.12: Eyes centre detection
Image Courtesy: [21]

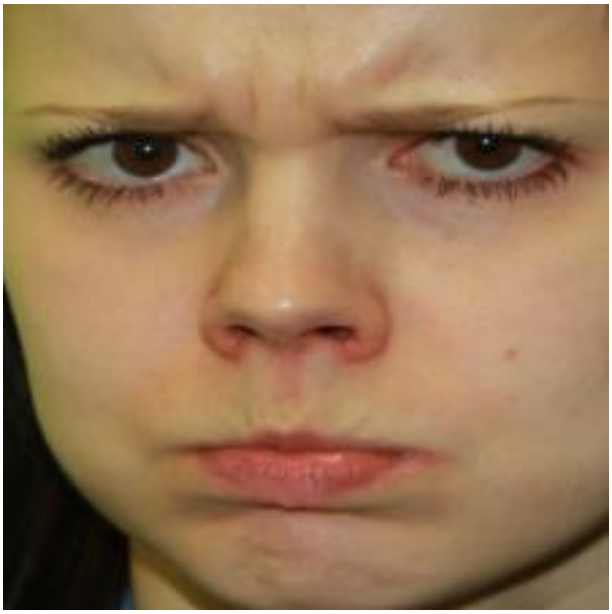


Fig. 4.13: Registered image based on eyes
Image Courtesy: [21]

Chapter 5

Integration

5.1 Overview

As described previously, in the Introduction the overall work is divided into several independent modules of which this report describes in detail work done on Scene Recognition, Emotion Recognition and integration of all the modules, and the other complementary report [4] describes about Character and Avtivity Recognition. But overall the task of synthetic video generation requires these modules to work together, this section describes as to how the integration of these modules is done.

5.2 Description

The very first criteria to be ensured for modules was to split the video into images (referred as frames). Each of the video was sampled at the original frame rate of about 23 frames per second (fps). This resulted in an average of 31,000 frames per episode and hence 24*31,000 = 7,44,000 frames from a single season of the series. Running each these modules independently on such large number of images at test time wouldn't be a good idea because for each different case this would mean redundant work. Therefore, every module which did not require context of a 'query' was applied on these frames to store the results beforehand. For instance we have had applied the character recognition pipeline (Chapter 3) on each of these frames and stored the results in form of dictionary of key-value pairs. Figure 5.1 shows such

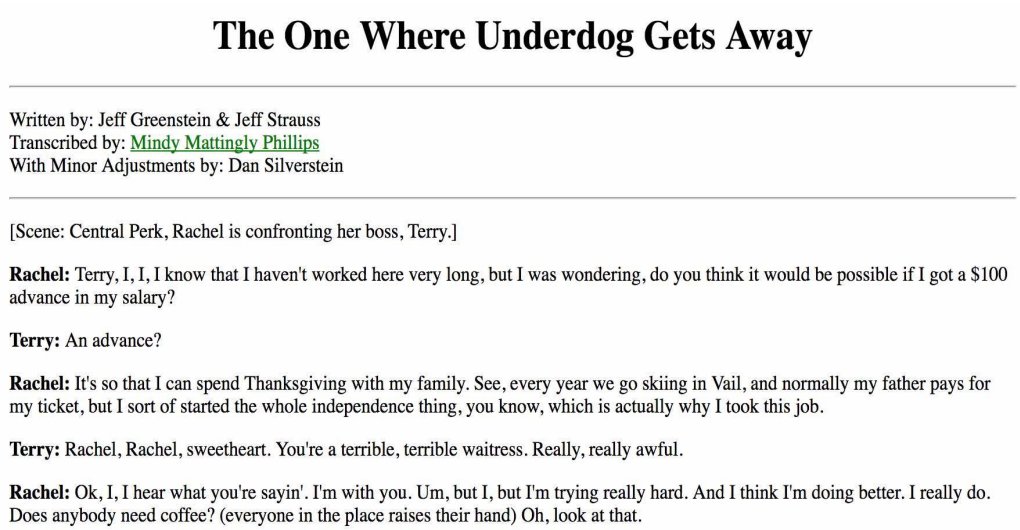
a dictionary of 13th Episode of 7th season (which is used for the project).

Fig. 5.1: Dictionary showing key-value pairs as results from character recognition module

The key is the frame name (number) and the value is a list consisting of number of faces in the frame followed by the coordinates of face and the probability of that face for each of the character classes. Similar type of results were obtained and stored after running other such modules on each of the frames.

Each of the modules have had been designed to work on images rather than video, to ensure independence of a frame from another. However, this results in loss of temporal context in the video, we are trying to generate. This temporal context is brought while integrating these modules. Before getting the sample script we make short video segments of about 1-2 second. These videos are generated by using the previously stored results of modules. For instance, for generating video of Chandler speaking at Central Perk, We first consider all the frames of Chandler and then all those continuous segments of frames of 1 second which are tagged as speaking; finally we sort by decreasing order of probability of location being central perk and pick top 'n' number of videos. These short videos are generated corresponding to each such query - Person, Speaking/non-Speaking, Location.

For generating the video of an episode, a sample script 5.2 is given, first we parse this script and split into scenes and generate the synthetic videos scene wise. Further, in parsing



The One Where Underdog Gets Away

Written by: Jeff Greenstein & Jeff Strauss
Transcribed by: Mindy Mattingly Phillips
With Minor Adjustments by: Dan Silverstein

[Scene: Central Perk, Rachel is confronting her boss, Terry.]

Rachel: Terry, I, I, I know that I haven't worked here very long, but I was wondering, do you think it would be possible if I got a $100 advance in my salary?

Terry: An advance?

Rachel: It's so that I can spend Thanksgiving with my family. See, every year we go skiing in Vail, and normally my father pays for my ticket, but I sort of started the whole independence thing, you know, which is actually why I took this job.

Terry: Rachel, Rachel, sweetheart. You're a terrible, terrible waitress. Really, really awful.

Rachel: Ok, I, I hear what you're sayin'. I'm with you. Um, but I, but I'm trying really hard. And I think I'm doing better. I really do. Does anybody need coffee? (everyone in the place raises their hand) Oh, look at that.

Fig. 5.2: Sample script used to generate videos

we split the scene into segments (dialogs) where each segment of time duration approximated by amount of words in the dialog. Now this segment is similar to the query of our beforehand generated short videos having info of character, action and location. So to justify this we pick up a small video and the next video is picked such that the disparity (measured by normalized cross correlation) between last frame of already picked video and first frame the video being considered is minimized. This is then continued till we fill the expected time duration of that segment with these short 1-2 second videos. Before adding video of next segment we add a non-speaking short video of multiple characters present in that scene to make the video more natural. Further the first video of next segment is picked by disparity minimization and similar process is repeated. Instead of just considering only the video which minimizes disparity we make it stochastic and pick the next video with probability inversely proportional to the disparity; this gives us multiple videos for a scene and hence multiple videos for an episode.

References

[1] M. Ajmal, M. H. Ashraf, M. Shakir, Y. Abbas, and F. A. Shah, "Video summarization: Techniques and classification," in *Proceedings of the 2012 International Conference on Computer Vision and Graphics*, ser. ICCVG'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 1–13. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33564-8_1

[2] MIT. (2016, Oct.) Mit places database. [Online]. Available: http://places.csail.mit.edu/

[3] F. Chollet. (2016, Aug.) Building powerful image classification models using very little data. [Online]. Available: https://blog.keras.io/building-powerful-image-classification-models-using-very-little-data.html

[4] Paawan Mukker, "Synthetic video generation - character and activity recognition," MTP Report, Department of Computer Science and Engineering, IIT Mandi, Mandi, May 2017.

[5] Wikipedia. (2016, Aug.) Friends. [Online]. Available: https://en.wikipedia.org/wiki/Friends

[6] M. Ajmal, M. H. Ashraf, M. Shakir, Y. Abbas, and F. A. Shah, "Video summarization: Techniques and classification," in *Proceedings of the 2012 International Conference on Computer Vision and Graphics*, ser. ICCVG'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 1–13. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-33564-8_1

[7] J. Johnson, A. Karpathy, and L. Fei-Fei, "Densecap: Fully convolutional localization networks for dense captioning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[9] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[10] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014. [Online]. Available: http://arxiv.org/abs/1409.1556

[11] L. Wang, S. Guo, W. Huang, and Y. Qiao, "Places205-vggnet models for scene recognition," *CoRR*, vol. abs/1508.01667, 2015. [Online]. Available: http://arxiv.org/abs/1508.01667

[12] C. A. Corneanu, M. O. Simn, J. F. Cohn, and S. E. Guerrero, "Survey on rgb, 3d, thermal, and multimodal approaches for facial expression recognition: History, trends, and affect-related applications," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 8, pp. 1548–1568, Aug 2016.

[13] G. Levi and T. Hassner, "Emotion recognition in the wild via convolutional neural networks and mapped binary patterns," in *Proc. ACM International Conference on Multimodal Interaction (ICMI)*, November 2015. [Online]. Available: http://www.openu.ac.il/home/hassner/projects/cnn_emotions

[14] V. Struc and N. Pavevsic, "The complete gabor-fisher classifier for robust face recognition," *EURASIP Advances in Signal Processing*, vol. 2010, p. 26, 2010.

[15] ImageNet. (2016, Aug.) Imagenet database. [Online]. Available: http://image-net.org/

[16] B. A. J. A.Torralba and A. Oliva, "Learning deep features for scene recognition using places database," *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[17] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1 - Volume 01*, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2005.177

[18] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[19] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 1867–1874. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.241

[20] B. Amos, B. Ludwiczuk, and M. Satyanarayanan, "Openface: A general-purpose face recognition library with mobile applications," CMU-CS-16-118, CMU School of Computer Science, Tech. Rep., 2016.

[21] Codalab. (2017) Emotion challenge. [Online]. Available: https://competitions.codalab.org/

[22] P. J. H. Prof. Matti Pietikainen. (2017) Image and video description with local binary pattern variants. [Online]. Available: http://www.ee.oulu.fi/research/imag/mvg/files/pdf/CVPR-tutorial-final.pdf

[23] VLFeat. (2017) Hog features. [Online]. Available: http://www.vlfeat.org/overview/hog.html

[24] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998. [Online]. Available: http://dx.doi.org/10.1023/A:1009715923555

[25] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.