ATTENDANCE MARKING SYSTEM USING IMAGE RECOGNITION

Professor : Sanjay Srivastava
Team :
Kunal Suthar(201401131)
Nirav Patel(201401179)
Maulik Limbadiya(201401189)

Introduction

Our proposed solution attempts to mark attendance of students during lecture hours by using multiple cameras at multiple places, which would take photos of the assigned portion of the class at multiple time intervals. These photos are stored at a server, where the face detection and recognition process commences.

Technology used

- **OpenCV** : An open-source library which allows us to use the HAAR cascade which is used for face detection and pre-processing the image.
- **Keras :** An open source neural network based library used for image recognition.
- **Python 2.7**

Implementation Steps

- Creation of database and training the convolutional neural network.
- Image acquisition from real environment in the form of frames.
- Face detection from the frames.
- Face recognition and testing of the neural net.
- Finding adequate image size and quality for achieving better detection and recognition accuracy.

Face Detection

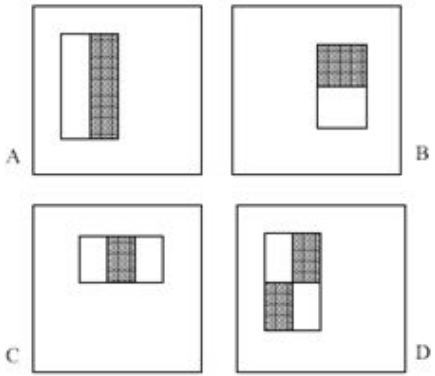
Object Detection using Haar feature-based cascade classifiers is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. Then we need to extract **features** from it.

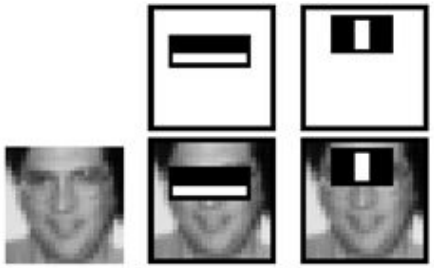
Four basic types of features:

– They are easy to calculate.

– The white areas are subtracted from the black ones.

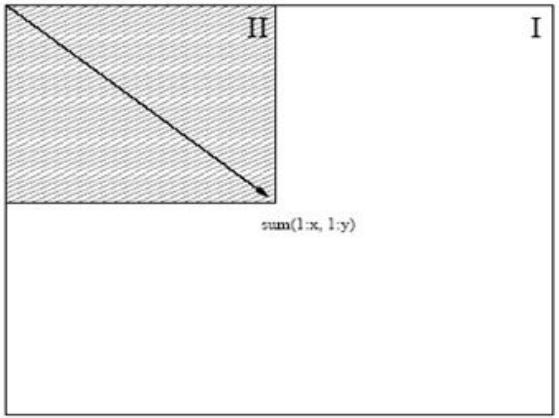
– A special representation of the sample called the **integral image** makes feature extraction faster.





Integral Images

- Summed area tables



- A representation that means any rectangle’s area can be calculated in four indexes to the integral image

Feature Extraction

1. Features are extracted from sub windows of an sample image.

–The base size for a sub window is 24 by 24 pixels.

- In a 24 pixel by 24 pixel sub window there are 180,000 possible features to be calculated.

2. What is the end result of feature extraction?

–A lot of data!

- This is called overfitting and the amount of data must be reduced.
- Overfitting can be compensated to an extent by logical elimination.

Feature/Classifier Evaluation

- Using AdaBoost the number of features is dramatically reduced.

–A simple algorithm that selects one feature at a time and assigns weights to the feature. It helps in combining multiple “weak classifiers” and thus producing a strong classifier.

$$h(x) = \begin{cases} 1 & \text{if } \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & \text{otherwise} \end{cases}$$

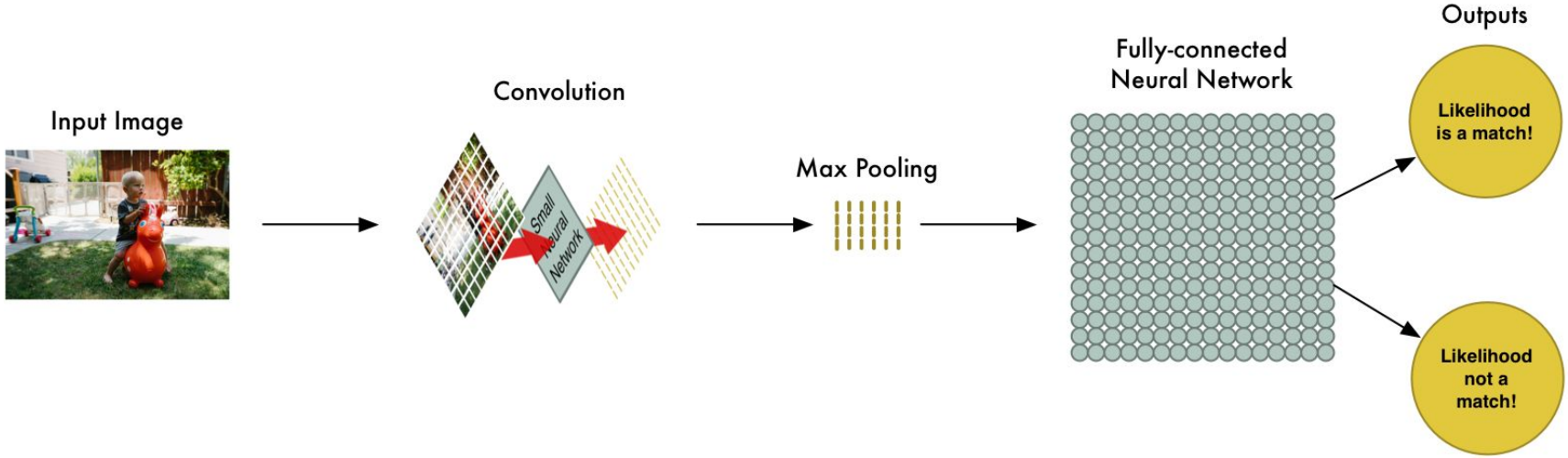
- Thus, these features helps us in face detection.

Face Recognition

We can train a neural network to identify images. But the problem faced in the image recognition is that the image is not always centered, as we had in our training data. We can use the concept of a sliding window, but it's inefficient. Instead, we introduce more data where the face can be anywhere in the image. We add more layers in the neural network. But, this is also inefficient. So, instead of training neural network with more number of data tuple, we increase the network size itself to reduce computation.

Working of convolution

Similar to a sliding window, we pass a window through an image and save the result of each window as a separate image. Feed each image in the small neural network. Image which gives the best classification results are selected. Array size is reduced. To reduce it further, we do max pooling. We look at the 2*2 window and the image with the highest weight among these are selected. And finally, it is applied to neural network.



Results

Face Detection : For images of good quality, the face detection algorithm provides pretty high accuracy. Following are some prerequisite constraints :

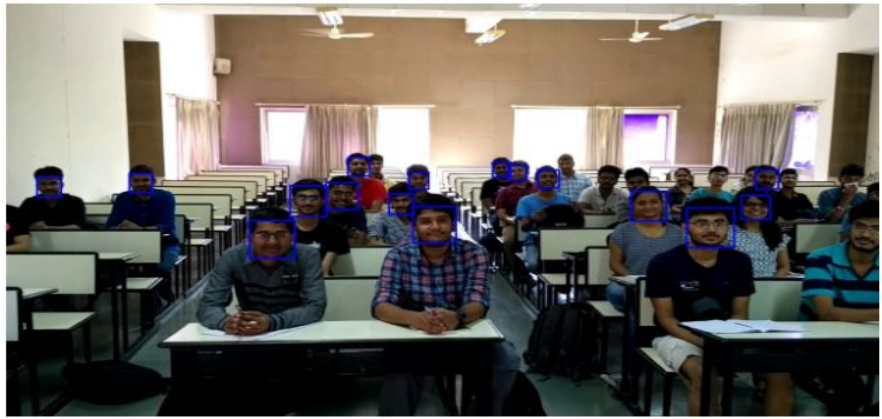
- The size of faces in images should be of at least 25*25 resolution.
- Blurred faces do get detected but cannot recognized. So blurred images are useless.
- The images shouldn't be too dark.
- Side faces aren't detected.

The results of face detection algorithm that we ran on a video recorded in a regular class are shown below. The given results are for a 30 fps video from which a frame from every 5 frames is extracted for face detection.

Following are two examples where some students who are not detected in first image get detected in the second image. So, a stream of such images covers almost all of the students.

FRAME-1

FRAME-2



FRAME-3

FRAME-4

The undetected faces in frames 1 & 3 gets detected in the subsequent frames 2 & 4

Face Recognition Results:

Following are the results for face recognition which we ran on an image database that we downloaded from internet particularly for face recognition. For each person around 30 images were available to train the model. Each image had a little variation like angle, sunglasses, facial expression. And because the quality of the images of given database was good and the total number of distinct persons were less, the accuracy is pretty high for only 10 epochs. For a fairly large amount of class, the total number of distinct student increases and so the number epochs to train the model increases too. This also gives an idea about the data collection process for the given system.

```
Train on 1255 samples, validate on 539 samples
Epoch 1/10
1255/1255 [==============================] - 1s - loss: 1.6321 - acc: 0.5554 - val_loss: 0.4641 - val_acc: 0.9703
Epoch 2/10
1255/1255 [==============================] - 1s - loss: 0.2750 - acc: 0.9538 - val_loss: 0.1077 - val_acc: 0.9870
Epoch 3/10
1255/1255 [==============================] - 1s - loss: 0.0876 - acc: 0.9865 - val_loss: 0.0758 - val_acc: 0.9833
Epoch 4/10
1255/1255 [==============================] - 1s - loss: 0.0596 - acc: 0.9928 - val_loss: 0.0500 - val_acc: 0.9963
Epoch 5/10
1255/1255 [==============================] - 1s - loss: 0.0332 - acc: 0.9912 - val_loss: 0.0763 - val_acc: 0.9759
Epoch 6/10
1255/1255 [==============================] - 1s - loss: 0.0341 - acc: 0.9912 - val_loss: 0.0464 - val_acc: 0.9963
Epoch 7/10
1255/1255 [==============================] - 1s - loss: 0.0413 - acc: 0.9896 - val_loss: 0.0515 - val_acc: 0.9907
Epoch 8/10
1255/1255 [==============================] - 1s - loss: 0.0135 - acc: 0.9960 - val_loss: 0.0482 - val_acc: 0.9963
Epoch 9/10
1255/1255 [==============================] - 1s - loss: 0.0084 - acc: 0.9992 - val_loss: 0.0459 - val_acc: 0.9963
Epoch 10/10
1255/1255 [==============================] - 1s - loss: 0.0055 - acc: 0.9992 - val_loss: 0.0448 - val_acc: 0.9963
352/539 [==================>...........] - ETA: 0s0.044755568255
0.996289424861
(cv) maulik@maulik-HP-Pavilion-15-Notebook-PC:~/RI$
```

Conclusion/Future Extension

- What we did is just the groundwork for implementing the attendance system i.e., implementing the software part of it.
- A network of cameras and server is to be setup to actually implement the system.
- Deployability in an actual class is yet to be tested.

References

1. An Automatic Attendance System Using Image processing by Aziza Ahmedi, Dr Suvarna Nandyal in The International Journal Of Engineering And Science (IJES).
2. Real-Time Face Detection Using Gentle AdaBoost Algorithm and Nesting Cascade Structure by Jian-qing Zhu and Can-hui Cai at 2012 IEEE International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS 2012) November 4-7, 2012.
3. Rapid Object Detection using a Boosted Cascade of Simple Features by Paul Viola and Michael Jones at 2001 IEEE.
4. Real Time Face Recognition Using Adaboost Improved Fast PCA Algorithm. International Journal of Artificial Intelligence & Applications (IJAIA), Vol.2, No.3, July 2011
5. Medium Articles for the convolutional neural network part: https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721