Speaker Classification in Images

Assignment 2 – ELL888

IIT Delhi

Anish Mahishi – 2014MT10585
Adarsh Onkar – 2014MT10582

Date – 17/04/2018

Problem Statement

Given a set of videos for 6 different speakers as training data, construct and train a Neural Network Model to classify still images as one of the given speakers or Noise. Speakers are as follows:

1. Sandeep Maheshwari – Motivational Speaker
2. Sadhguru Jaggi Vasudev – An Indian Yogi and Mystic
3. Sorabh Pant – Stand-up Comedian
4. Atul Khatri – Stand-up Comedian
5. Shailendra Kumar – Guitar Tutor
6. Raman Kalyan – Flautist

Training Data

Training data comprises of wide-ranging images extracted from 5 videos of each speaker. In particular, we use around 500 images of each speaker for training. The Training Set is generated through the following procedure:

1. Image Extraction

- We used Windows Movie Maker for splitting and joining parts of the video.
- Any clip containing only the speaker or the speaker with some other people is considered to be relevant.
- Any clip not containing the speaker at all is considered to be Noise.
- As a result of the above, we get 6 processed videos containing only the speakers and no noise.

2. Noise Extraction

- Noise is extracted from the same videos from which speakers are extracted and we get a separate video containing only Noise.

3. Image Sampling

- The videos generated as a result of previous steps gives us 7 different videos at a frame rate of 30 fps. We take 1/90 frames (,i.e. 1 frame per 3 secs) assuming that at such a sampling rate, we can get less redundancy between different frames of a speaker in the training data and we can cover sufficient variation for each speaker through this . Thus we get around 500 frames for each speaker also maintaining the balance in the training data set

Pre-Processing

Before feeding the images to train the model, we apply the following pre-processing steps:

- Resizing

  We tried training the model on original images (without resizing) and faced the following issues:

  - The virtual machine crashed due to *Memory Limit Exception* and *Resource Exhaustion Error* after some time interval.
  - The overall training required a lot of time.

  Considering the above limitations, we resized all the images in the training data set into a fixed size of 150x120. We observed that the basic visual features of all the speakers is conserved even after resizing.

After resizing the image, we tried 3 different types of pre-processing and tested them separately on the testing dataset to obtain accuracies. They are as under:

1. No pre-processing:

   First of all we tried training the model on the entire 150x120 image to check what the network learns, given we do not give it any prior knowledge that it has to classify the person in the image. We expected the CNN to learn features like the background under such a set-up.

2. YOLO:

   We used *lightnet* which is a python wrapper over *darknet* and includes the implementation of YOLO. This model is trained over the Coco dataset for object-detection. For our purpose, we use it for person-detection. After detecting persons in the image, we count the number of persons.

   For frames belonging to one of the six speakers:

- 1 person: Use this frame for training. Crop the person, use the cropped image for training.
- Several persons: Discard this frame

   For frames belonging to Noise:

- No person: Take the entire image for training.
- One or more persons in the frame: Crop each person and take each such cropped image for training. Apart from this, also take the entire image for training. So a single noise image with n people will result into (n+1) training samples for noise.

After cropping image after applying YOLO, we resize the cropped image back to 150x120 size to maintain consistency in frame sizes.

3. Face-Recognition:

We use *face_recognition 1.2.2* which is a *dlib's* implementation of state-of-the-art face recognition system trained over LFW Face Database. After detecting faces, we count the number of faces.

For frames belonging to one of the six speakers:

- 1 face: Use this frame for training. Crop the face, use the cropped image for training.
- Several persons: Discard this frame

For frames belonging to Noise:

- No face: Take the entire image for training.
- One or more faces in the frame: Crop each face and take each such cropped image for training. Apart from this, also take the entire image for training. So a single noise image with n faces will result into (n+1) training samples for noise.

Similar to what we did in the case of YOLO, we again resize all cropped images back to 150x120 size.

Architecture

We tried different CNN architectures namely, VGGNet, ResNet and InceptionNet. The performance of VGGNet was the best for testing data generated by us from videos not used for training. So, we proceed to use the *VGG16* for our purpose.

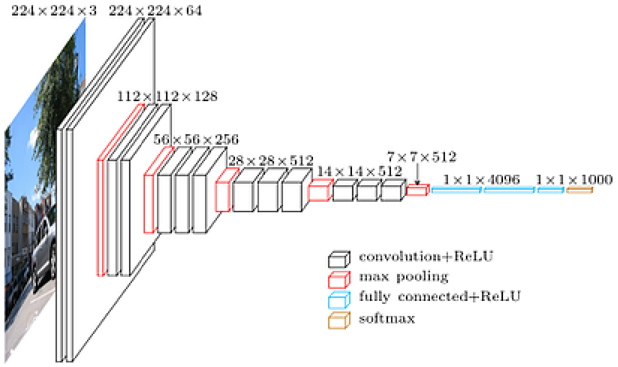

Image Reference: https://www.techkingdom.org

Results and Interpretations

1. No pre-processing:

The model seems to learn the background of the speakers if training is done on entire image without extracting faces or persons. Accuracy = 10.4%

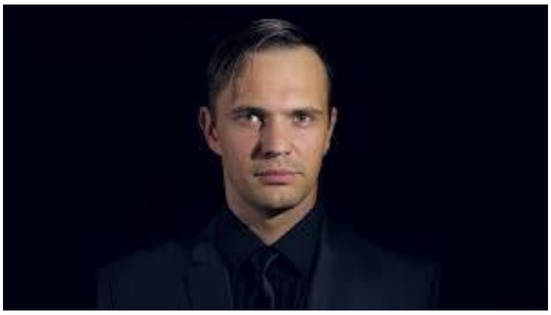

Figure 1: Classified as Sadhguru

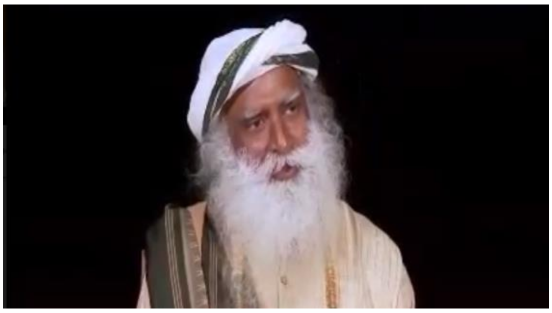

Figure 2: Classified as Sadhguru

2. With YOLO:

We tried to interpret what the model learns in case single person images are given to the model after applying YOLO. We did by doing gradient ascend on white noise for each of the speakers. The corresponding images are as under:

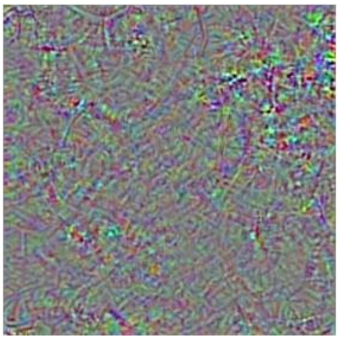

Figure 1: Sandeep Maheshwari

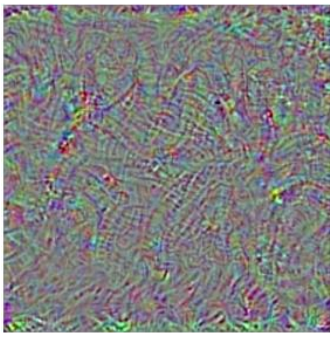

Figure 2: Sadhguru

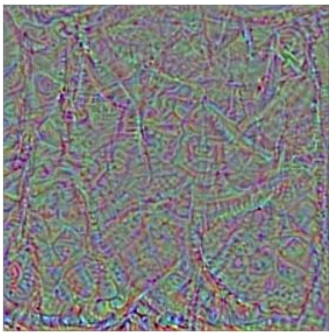

Figure 3: Atul Khatri

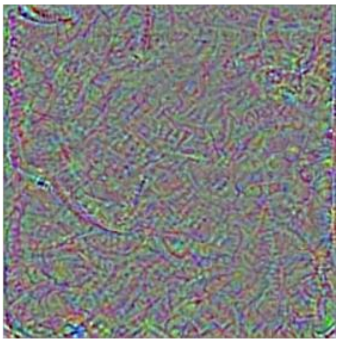

Figure 4: Sorabh Pant

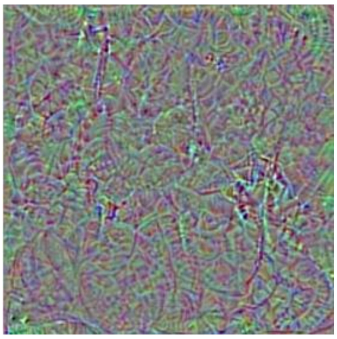

Figure 5: Flute Raman

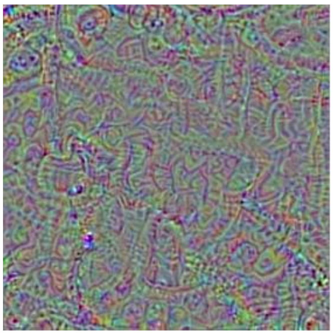

Figure 6: Shailendra

Accuracy = 57.5%

3. With Face-Recognition:

Using gradient ascend for face images, we get the following images:

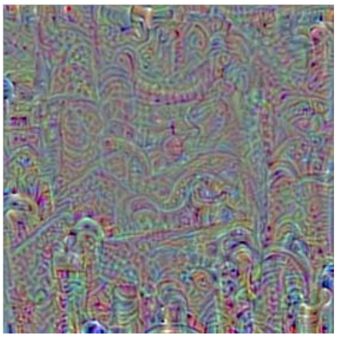

Figure 1: Sadhguru

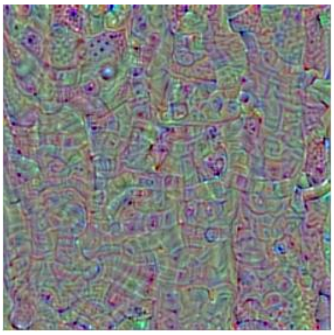

Figure 2: Sorabh Pant

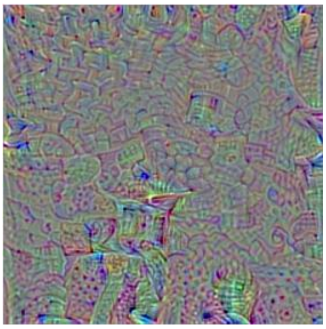

Figure 3: Flute Raman

Accuracy: 46% (Can be improved further as explained in Conclusion.4)

**All accuracies are on initial testing dataset (not the latest one)*

Conclusion:

1. VGGNet is performing better in the task as it has less no. of layers compared to InceptionNet and ResNet.

2. When no pre-processing is used, with limited training data-set where the background for a speaker is in general consistent, the network learns the background for the speakers and does not perform well on testing data.
3. The gradient ascend images does not show any general trend for YOLO processed images and the face-recognized images. What it means is that we should refrain from our natural tendency to anthropomorphize them and believe that they "understand", say, the concept of a bald man, or the appearance of beard, just because they are able to classify these objects with some accuracy.
4. Face recognition was supposed to perform better but the face detection model was slow which prevented us from fine tuning the model.
5. Like any other deep learning model, a more varied dataset will be better for this task. For our case, due to computational complexities and less variation among videos, even with high training accuracy, testing accuracy was not significant. A larger training set will probably make the model more efficient.

References

**References for the code sections used are commented in the code itself.*

**References for theory is provided as hyper-links in the theory itself*