**Title**: Intelligent Surveillance

**Authors**: Murat Ozan Aydin, Rajesh Chodavarapu, Sercan Demir, Bilijana Jovanova, Sean Liguez

**Project Definition:** This project presents an implementation of a data model which detects the features of the human face using Deep Learning (DL) techniques such as Convolutional Neural Networks (CNN). The goal is to identify human faces obtained from any image source such as Closed-Circuit Television (CCTV) system with a specific application to Intelligent Surveillance.

The Labeled Faces in the Wild (LFW) image dataset has 13,000 facial images from people of different areas of the media. From the entire dataset 80% is used for training the model and 20% is used to test for the model accuracy. The target label is the person’s name. We will assume that all of these people in the dataset are potential suspects or targets which the model will train on to recognize with high degree of accuracy.

The hypothesized application is to use the trained model to trigger an alert once the target has been identified in another image source of interest, such as CCTV. Thus, the use cases of this Intelligent Surveillance system can include criminal investigations, biometrics identifications, and surveillance systems.

The scope of our project is only to build the image classification model of all suspects. We will not be joining the output of our model to an actual CCTV data since we cannot acquire it. However, we wanted to present the overall idea and how the model can be used in a surveillance system.

**Outcome:** Applydeep learning techniques to identify people from a given number of images provided to the training model.[[1]](#footnote-1) Vis-cs.umass.edu. (2018). *LFW Face Database : Main*.Available at: <http://vis-www.cs.umass.edu/lfw/>.

**Dataset:** The image data can be found at [**http://vis-www.cs.umass.edu/lfw/**](http://vis-www.cs.umass.edu/lfw/)**.** The LFW dataset contains more than 13,000 images of faces from different areas of interest. Each face picture has been labeled. This consisted of 5,749 people who had different number of images in the data, some >500 and some only one.

I. Introduction

Today, surveillance is increasingly more important because it can be used for threat detection, intruder detection, or to detect abnormal activities. Our project was to **develop a small surveillance system in which we have known people (targets) that the model has been previously trained on. By using this model, we can successfully identify these targets from any image data.**

According to Arc Map, “Image classification refers to the task of extracting information classes from a multiband raster image. The resulting raster from image classification can be used to create thematic maps. Depending on the interaction between the analyst and the computer during classification, there are two types of classification: supervised and unsupervised.” Supervised image classification uses facial embeddings such as: eyes, nose, mouth and ears obtained from training samples to find a face an image and match it to previously labeled data points provided to the system. It is important to make this system smart enough to identify suspects and to alert the security team to help them prevent any abnormal activities or to catch an identified criminal. Furthermore, having a manageable system that can easily learn simple models will create new tools that help spot crime, accidents and industry hazards. Today, we will explore the LFW dataset, our project methodology and setup, the results we gathered, and the implications of this data.

II. Project Methodology

Using Artificial Neural Networks (ANN), we develop a model that classifies the person based upon a given training data set. For this purpose, we are using deep learning models such as CNN and open source image processing systems such as OpenCV. For this problem, we chose the LFW dataset (Vis-cs.umass.edu, 2018) to train the model. The image data provided has of 13,000 facial images from people of different areas of media. Each person has been labeled with a his or her name person. We hypothesize that the people in the dataset are the suspects. To train, we feed an input image and associate a name to it. The model building consists of two phases: construction and execution. In the following section we will describe the process:

1. *Model Building – Construction Phase - Topology*
   1. Data acquisition: Download the data set from LFW ([**http://vis-www.cs.umass.edu/lfw/**](http://vis-www.cs.umass.edu/lfw/))
   2. Data cleaning: Align the faces in the dataset
   3. Model topology: Build the Convolutional Neural Network topology
   4. Data splitting: Prepare the training data set- 80% (train 90% + validation 10%) and test dataset-20%
   5. Train Model: Use the training Data to build the model
   6. Model validation: Test the model accuracy using test dataset
   7. Repeat c-f until desired accuracy is achieved
2. *Face Identification – Execution Phase*
3. Capture images from any source (CCTV, Cameras, Drones)
4. Resize the input images if they are too big (ex. 4k images). This may result in data loss.
5. Identify the faces in the image from step B.b
6. Align the identified faces, if the face cannot be aligned label it as “unidentified”
7. Reshape the images (convert to grayscale)
8. Use the model from the construction phase (A) to classify the face based on the labeled faces
9. Identify the face if the accuracy probability threshold exceeds 40%, otherwise they are “unknown”
10. Draw a rectangle on the original image outlining the face and show the label name (‘xxx’ or ‘unknown’)

*What tools did we use?*

We performed OS setup in Ubuntu and created our environment in Python 3 on the Jupyter platform hosted on OpenStack Cloud Servers. Several Python packages were used in this project. OpenCV2 was used primarily for the image processing, which enabled us to read, render, shrink and modify an image. TensorFlow was used to build our neural network. Keras was used in our neural network and training the model. Openface and Face\_recognition was used for facial recognition. Dlib was used as part of our face alignment process.

III. Project Results

The results from the facial recognition are as follows:



Figure 1: The relationship of number of images with the output and veracity of the dataset.

*Data Analytics Process:*

Our base model contains the entire cleaned and aligned training dataset (9,253 images of 1,676 people). To train we added an additional constraint that the labeled person needs to have at least two images in the data. The activation function is RELU (hidden layer) and SoftMax (output layer) and the optimizer ADAM was used a hidden layer structure of 13 layers. The model accuracy was very poor with 36%. We tried another activation function (Sigmoid) and added another hidden layer and achieved marginal improvement of only 1%. We concluded that adding more layers and changing the activation function is not enough to achieve better accuracy.

We outputted errors from the model and realized that the model performs bad when the number of images per person is low.

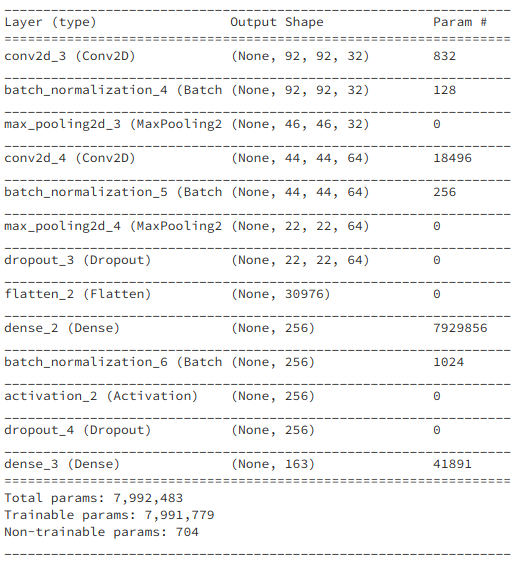
Thus, we restricted the model to include the labeled data which contains at least five training images per person. Total number of images retained after this constraint was 6,089, thus we removed 3,164 images. This second model was fit using RELU and SoftMax with ADAM optimizer and the first hidden layer structure. The model accuracy increased by 22% to 58%. To see if adding more complexity in the hidden layers will increase accuracy, we added an additional hidden layer. This increased the accuracy by 3% to 61% but computational time was also increased.

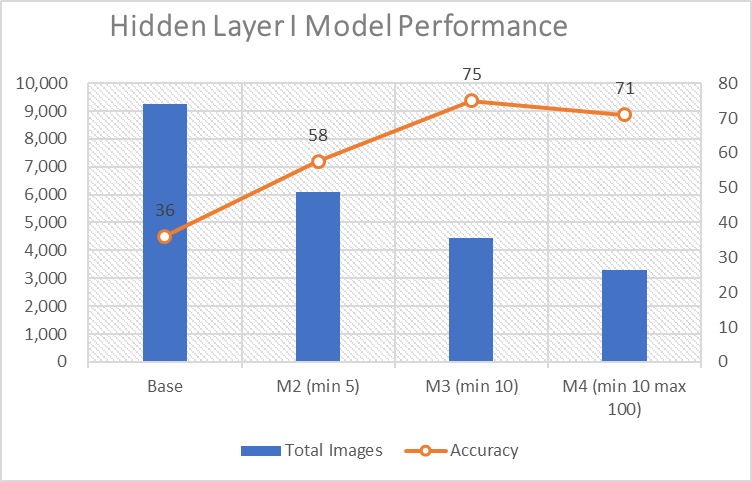
Even after restricting the data model to five images per person, we still had very imbalanced dataset and most of our errors were on targets (people) with low number of images. Thus, we decided to test whether further restriction in terms of number of images will improve the model.

The third model we fit restricted the number of images per person to 10. This resulted in a loss of 1,649 images. We used the same model specifications (RELU, SoftMax, ADAM) and first trained with the first hidden layer structure. The model accuracy increased significantly to 75%. Adding another hidden layer resulted in a loss 4%, thus we concluded that the added complexity is not needed.

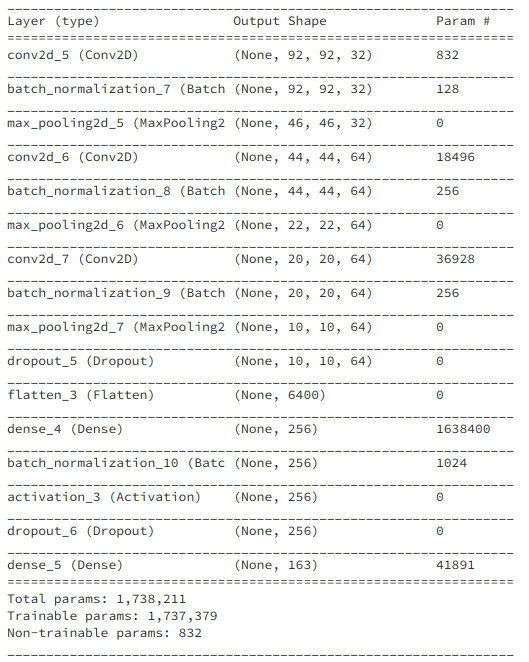
*Details:*

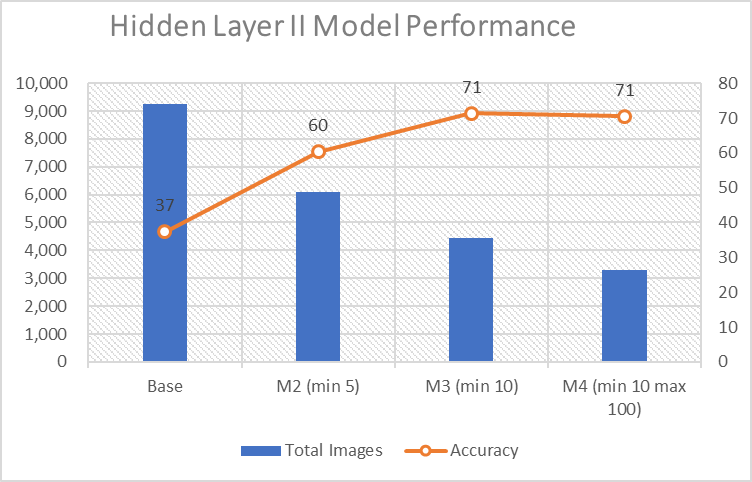
First Hidden Layer Structure:





Second Hidden Layer structure:





We tried adding more layers and tried different activation functions, but no improvement was made to the overall accuracy. We concluded that the highest accuracy with the existing data is the one achieved in our third model with minimum 10 images per person.

*Model improvement ideas:*

The best way to improve the model is to improve the quality of the data. We restricted the training data to a specific number of training images per person as shown in our Data Analytics Section. We also changed the topology parameters, trying different activation functions, optimizers, or adding more hidden layers. The best way we increased the accuracy of the model was by lowering the amount of training images. While this gives us the intended results in terms of increased accuracy, we are essentially removing a lot of data in our training phase. Data is scarce and in practicality it might not be a good idea to remove people, especially if those people are target suspects. Ideally, we would acquire more data per person to try to balance the dataset. This taught us is that in order to increase accuracy we need at least 10 images per person, thus the next step would be to up-sample the training data to balance the number of training images per person. By properly balancing the dataset, we can provide more images while increasing the accuracy of the data model.

IV. conclusion

From our findings, we confirmed the simple rule that data volume and veracity are paramount to the success of good modeling and data analytics. Challenges we faced included an unbalanced dataset, unaligned faces, computation time, and size variation in the images. With the computation power we have now, we built a model with an acceptable level of accuracy and we used this model to identify individuals from a group of people. Next steps we would consider are balancing the image data by increasing the number of images per person to at least 10 if not more. Due to time restriction we could not fully explore this idea. Other ideas we would consider is making this adapt the model recognize to emotions or applying the model to a live video.

V. Appendix

All charts, graphs, figures, and pictures

VI. Acknowledgements

Paul Rad.

VII. References

Opencv.org. (2018). *About - OpenCV library*. [online] Available at: <https://opencv.org/about.html> [Accessed 1 Dec. 2018].

(Opencv.org, 2018) – in text

Vis-cs.umass.edu. (2018). *LFW Face Database : Main*. [online] Available at: http://vis-www.cs.umass.edu/lfw/ [Accessed 1 Dec. 2018].

(Vis-cs.umass.edu, 2018)

Desktop.arcgis.com. (2018). *What is image classification?—ArcGIS Help | ArcGIS Desktop*. [online] Available at: http://desktop.arcgis.com/en/arcmap/latest/extensions/spatial-analyst/image-classification/what-is-image-classification-.htm [Accessed 1 Dec. 2018].

(Desktop.arcgis.com, 2018)

Footnotes format:

First, Last, Title (Place of publication: Publisher, Year of pub), page number

1. Vis-cs.umass.edu. (2018). *LFW Face Database : Main*. [online] Available at: http://vis-www.cs.umass.edu/lfw/ [Accessed 1 Dec. 2018]. [↑](#footnote-ref-1)