# Capstone Project Final Report

## Emotions Detection - Classification

Artificial Intelligence & Machine Learning: Business Applications

Dr. Nolan Grieves
nolangrieves@gmail.com

## KEYWORDS

Machine Learning, Artificial Intelligence, Deep Learning, Convolutional Neural Networks, Emotions detection

## 1 EXECUTIVE SUMMARY

**Machine Learning and Artificial Intelligence algorithms have a wide range of applications including the ability to use Computer Vision to recognize specific features within images and determine if an image belongs to a predefined category. In this project I built several machine learning models using the programming language Python to determine if an image of a face was happy or not happy. The data set consisted of 48x48 pixel grayscale images including 4022 images for training the models and 424 images to test the models.**

**Five different types of machine learning algorithms were built including Logistic Regression, Random Forest, XGBoost, Artificial Neural Network (ANN), and Convolutional Neural Network (CNN) models. After tuning and optimization, the models achieved testing accuracy from ~70-80%. The CNN model was the best performing with a testing accuracy of ~81%. The structure of this CNN model (convolutional, pooling, and fully connected layers) can be used to detect facial emotions with >80% accuracy, but also could be trained to recognize and classify a wide range of images. However, the structure of this model should be further tested including the number of specific types of layers and their location as well as the hyperparameters used within the model. If further optimized this model could be able to detect a wider range of emotions more accurately, specific facial features, and possibly individuals.**

## 2 PROBLEM AND SOLUTION SUMMARY

The objective of the project was to build a machine learning algorithm that can identify if an image of a face was happy or not happy. In this section I give a report of the data, an overview of the different solutions applied, and present the final model.

### 2.1 Data Report

The name of the data set is **FER-2013** which is an open-source data set that was made publicly available for a Kaggle competition [4]. The data consists of 48x48 pixel grayscale images of faces. The faces have been automatically registered so that the face is more or less centered and occupies about the same amount of space in each image. In the original data set on Kaggle there were seven categories (angry, disgust, fear, happy, sad, surprise, neutral). The training set consists of 28,709 examples and the public test set consists of 3,589 examples. For the Great Learning Capstone Project only a part of the data set is taken to get better understanding and inference of CNNs. The original data set has been modified such that there are only two emotion categories (happy and not happy).

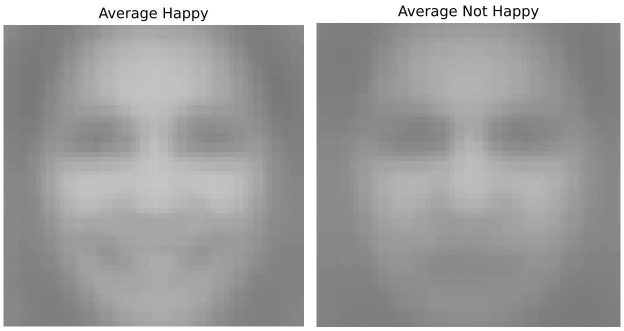

**Figure 1: *Left:* Average of all happy images. *Right:* Average of all not happy images.**

The data was likely taken from publicly available images and resized to center on the face and turned into grayscale as well. Several celebrities are in some of the images so they could come from movies or publicly available pictures from movies or tv shows. All of the images are in .jpg format. The data is only slightly imbalanced with 2000 happy images and 2022 not happy images for the training data set. The test data set is much smaller and again only slightly imbalanced with 200 happy images and 224 not happy images.

The data was first pre-processed by converting the jpeg files to arrays of pixel values, flattened, and saved as csv files. The flattened data arrays were used for the Logistic Regression, Random Forest, and XGBoost models, but the CNN models required the arrays to remain in their 48x48 pixel format. Before the data was inputted into the machine learning algorithms, it was split into train and test data sets, labeled, and scaled by 255 (the max pixel value).

Various Exploratory Data Analyses (EDA) were used including finding the average images as seen in Figure 1, creating difference and variability images, changing image format (e.g., RGB, HSV, Gaussian blurring, image size), and using a Canny edge detector. This EDA found that average images showed the smile evident in happy images and upper cheek more prominent. The difference image showed chin, forehead, and upper cheeks to be most different between happy and not happy images. The final data inputted in to the model was not manipulated beyond normalizing in order to keep the models simple; however this could be further tested during model optimization.

*Great Learning, AIML: Business Applications, 15 August 2022*
.

## 2.2 Machine Learning Models

Various models besides the final algorithm were tested. These models gave performances between ~68-74% on testing data.

- A **Logisitc Regression** model was optimized using a 'saga' solver and a threshold of 0.502 from the AUC-ROC curve [3]. The final performance gives an accuracy of 0.86201 on train data but only 0.68160 on test data.
- A **Random Forest** model was optimized using a randomized cross validation (CV) grid to tune hyperparameters. The final performance gives an accuracy of 0.733 on test data, but the model is over-fitting the data as training performance has an accuracy of 0.994.
- An **XGBoost** model was also optimized by tuning hyperparameters with a randomized CV grid. This model also over-fit the data with an accuracy of 1 on the training set and 0.74292 on the testing set.
- A basic **Artificial Neural Network (ANN)** was implemented and optimized using randomized CV grid to tune the hyperparameters batch size, learning rate, and dropout rate. Early stopping was tested on the ANN and a threshold was optimized using a AUC-ROC curve. The optimized ANN model gives an accuracy of 0.696 on test data.

The final and best performing model was a **Convolutional Neural Network (CNN)**. CNNs are more often utilized for classification and computer vision tasks compared to other machine learning models [1] including ANNs as they perform better at capturing relevant features and ignore spatial and translational transformations. CNNs use filters to reduce the dimensionnality of an image and extract only important information and also require much less trainable parameters in comparison to ANNs.

A base CNN model was built using three sets of convolution (128, 64, 32 filters) max pooling, and batch normalization layers. Then a flattening layer, two sets of dense (64 and 32 neurons) and dropout layers, and an output layer was added. This base CNN model obtained 0.816 accuracy on test data. A second CNN model was built using transfer learning from VGG16 [2], which obtained 0.694 accuracy on test data. Finally the CNN model hyperparameters were tuned using a Keras Classifier including the number of neurons in the hidden layers, the number of epochs, the dropout size, and the batch size. This tuned CNN model obtained approximately 0.81 to 0.83 accuracy on the test data depending on the run. This CNN model was also tested on the full Kaggle data set with seven emotions and achieved 0.546 accuracy on the test data set.

A summary of the different performances for each model is presented in Figure 2. The CNN model gives the best accuracy on the testing data set and is the preferred final solution. As mentioned earlier CNN models have advantages over other machine learning algorithms for computer vision as they are better at capturing relevant features and ignore spatial and translational transformations. The CNN model should be used as a final solution for this problem; however, as discussed in the next section, the specific structure and hyperparameters of the CNN model should be further explored to increase performance. Specifically the CNN model is still overfitting the data as training performance is much higher than testing performance, so more dropout layers could increase performance.

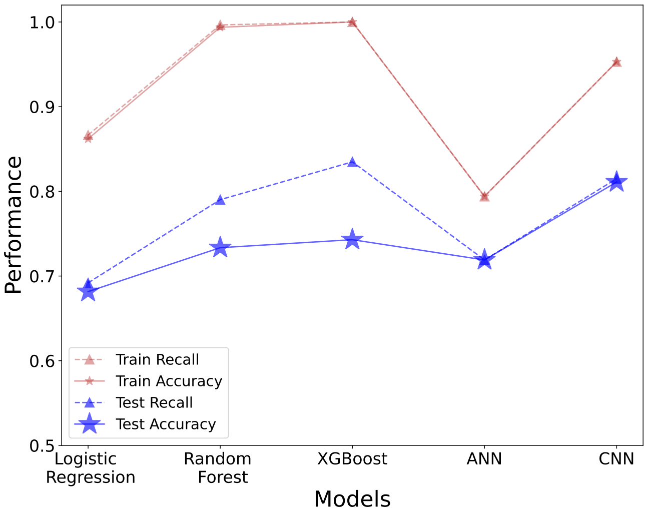

**Figure 2: Summary of performance for various Machine Learning models.**

## 3 RECOMMENDATIONS FOR IMPLEMENTATION

The CNN model should be used for the final solution to identify emotion (happy or not happy) in the images. The CNN had the best test accuracy of the investigated algorithms, and CNN models in general have been shown to outperform other machine learning models in computer vision tasks. The CNN model can obtain >80% accuracy on testing data, and takes advantage of known differences identified between the images. This includes insights from the EDA, which showed the smile to be evident in happy images and the upper cheek more prominent, as well as prominent differences between the two classes including the chin, forehead, and cheeks.

The CNN model was applied to the full Kaggle data set with seven different emotions and obtained 0.546 accuracy on testing data, which shows there is still much room for improvement to the model. Notably the structure of the model should be further tested including the number of convolutional layers, pooling layers, dropout layers, and dense layers, as well as where these layers are placed. The hyperparameters should be further tested including the number of filters in the convolutional layers, number of neurons in the dense layers, dropout rate, number of epochs, and batch size. Data augmentation could be tested including image rotation. Finally, the model is overfitting the data and techniques such as more dropout layers should be tested. If the model is improved enough it could be applied to detect emotion in general, specific facial features, and possibly the recognition of specific people.

## REFERENCES

[1] IBM Cloud Education. 2020. Convolutional Neural Networks. https://www.ibm.com/cloud/learn/convolutional-neural-networks
[2] Keras. 2022. VGG16 and VGG19. https://keras.io/api/applications/vgg/
[3] Sarang Narkhede. 2018. Understanding AUC - ROC Curve. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5
[4] Manas Sambare and Kaggle. 2013. Kaggle FER-2013. https://www.kaggle.com/datasets/msambare/fer2013