Computer Vision Application to Detect Emotions from CCTV Footage

Elizaveta German
elizaveta.german@studenti.unipd.it

Derek Sweet
derekallen.sweet@studenti.unipd.it

Johanna Weiss
johanna.weiss@studenti.unipd.it

1. Introduction

Using computer vision to detect emotions through facial detection has many potential real world applications for businesses. For example, businesses such as haunted houses make money by eliciting emotions of customers who pay to get scared for entertainment. If the haunted house is not scaring people, it is not delivering the service customers paid for. Computer vision can monitor the emotions of the customers when going through the attractions to identify issues with the attraction where people are not getting scared. This paper will explore the best models and pipelines necessary to build an application that can detect if customers are getting scared when going through a haunted attraction.

1.1. Challenges

Outside of the challenges typically faced when training emotional detection models from faces, this task presents three unique challenges to overcome.

(1) Customers faces will be difficult to see in CCTV security camera footage given the position of the cameras, and poor lighting conditions. Noisy images will be the biggest concern, so an experiment is conducted to compare the best method for denoising the images.

(2) Staged photos of actors pretending to be scared are used in the labeled data sets, however these expressions do not always match the reality of a truly scared customer. In order to overcome this difference, a pipeline needs to be created to allow the model to be fine-tuned using real footage of customers. The difficulty is that the data will be unlabeled, so an experiment is conducted to see the best way to automate the labeling of unlabeled data.

(3) People go through the attractions in groups, so multiple faces will need to be detected at one time in a single image. Two standard face detection algorithms are compared on simulated security footage data to determine which one will be better to detect and extract faces.

2. Datasets

For this application, the data used for training and experiments needs to reflect the type of images that will be captured by infrared CCTV cameras. All images will be greyscale and the resolution of the images will be converted to 48x48 pixels if they are not already that size. The main data used for training the models comes from two sources (FER-2013 [2] and CK+ [5]), both of which contain classified images with 7 facial expressions: 0=Angry, 1=Disgust, 2=Fear, 3=Happy, 4=Sad, 5=Surprise, 6=Neutral (see Figure 1). Even though the task is to identify people who are surprised, all available emotions in the data will be classified to be able to see if people are fearful or happy as well. The faces are cropped, centered and have equal space within each image. The two datasets combined contain 33,279 images split into three groups: train-29,494, test-3,678 and validation-3,678.

2.1. FER-2013

The largest source of facial expression data comes from Facial Expression Recognition (FER-2013) [2] which contains 32,298 grey-scale 48x48 images. The data was created by Pierre-Luc Carrier and Aaron Courville and preliminary versions of their data was provided on Kaggle for a competition.

2.2. CK+

The Extended Cohn-Kanade (CK+) dataset [5] obtained for this paper is much smaller with 981 images. The 48x48 facial images were captured in a lab and contain short temporal sequences of images. The focus of this paper will not leverage the temporal features of the data.

2.3. Bing Scraped Images

In additional the the classified data, a small amount of unclassified images (n = 95) are scraped from Bing Images to simulate what may be captured in the security cameras in the haunted attractions. The Bing query used for image scraping is 'scared haunted house reactions' which contains

Figure 1. Example of the 7 emotions for FER and CK

hundreds of images taken of customers at a haunted house (see Figure 2). This data is used to test the pipeline for self-training classification and evaluate the best face detection.

Figure 2. Example of the Bing image

2.4. Image Augmentation

Augmenting data is a popular way of generating more data for training models, or to make them more stable to varying data. In this project, the size of the data set is not problematic, so augmenting data is used primarily for the purpose of regularization and building a better-performing model. As images used for the emotion detection are of small size, augmenting them too much is making the critical feature undetectable. Therefore, the augmentation is limited to horizontal flip, shifting width and height to a factor of up to 0.1, a zoom range of up to 0.1, and a rotation range of 10.

In a separate experiment, noise perturbations are applied to the FER-2013 and CK+ data to experiment with denoise filtering methods. Two types of noise are applied to the images: (1) salt and pepper which turns 10% of the pixels black or white (2) coarse salt which replaces large rectangles of the image white salt noise (see Figure 3).

3. Method

To build and compare models that can classify emotions from photos correctly, the following methods are used. The models are initialized with Keras' callback options and in all models, convolutional neural networks are used. Additionally, transfer learning allows building more complex

Figure 3. Example of Salt and Pepper and Coarse Salt Noise

and pre-trained models. For denoising, autoencoders are used in comparison with a filter-based denoising approach. Considering the problem that image data often do not come with labels for training models, a self-training technique is used. Another important aspect is the detection of faces in images which is implemented with two different models. All results are evaluated based on the mean average precision and confusion matrix.

3.1. Initializing Models

Keras provides the option to define callbacks that perform actions at various stages of the training phase to make the process more efficient. For all of the following models, mainly three callbacks were used. The first one is early stopping with respect to the validation loss, which makes sure that the training of the algorithm stops once it converged to a certain value. If the validation loss function does not change significantly for five consecutive epochs the function is seen to be converged and the training is stopped. Moreover, the callback option of reducing the learning rate is used. This callback monitors the validation accuracy and if it does not show any improvement for two epochs, the learning rate is recalculated by a factor of 0.3. Finally, the modelCheckpoint class is used to save the model and weights when it has reached the best performance so far. The model can be loaded later to continue from the saved state.

For all models, the categorical cross entropy loss function is used, which is the most commonly used loss function for multiclass problems. It is based on the maximum likelihood and calculates the average difference between the actual and predicted probability distributions for predicting a class.

All models use Adam as an optimizer in the training phase. It is an adaptive optimizer that calculates learning rates for different parameters from estimates of first- and second-order moments of the gradients. It is usually more stable than using stochastic gradient descent and converges to the minima at a faster pace.

3.2. Convolutional Neural Networks

Convolutional Neural Networks are deep learning algorithms that are able to learn features from the input images

by applying different filters. The main goal of convolutional networks is to reduce input complexity so that it becomes easier to process and, simultaneously, not to lose features that are crucial for the prediction. The more convolutional layers are in the model, the more complex high-level features it can extract.

3.3. Transfer Learning

Transfer Learning is a method that allows building models by using patterns that have been learned on different problems. Pre-trained models are models trained on the large dataset (mostly, "ImageNet) which should ideally be similar to one that is used for the current problem. In case it does not match ("ImageNet" does not specify on faces), corresponding fine-tuning techniques should be applied. According to Figure 4 the problem of emotion recognition refers to Quadrant 1, therefore the entire model should be trained. Other options differ from the first by the amount of frozen (non-trainable) layers. Freezing layers help in terms of time consumption.

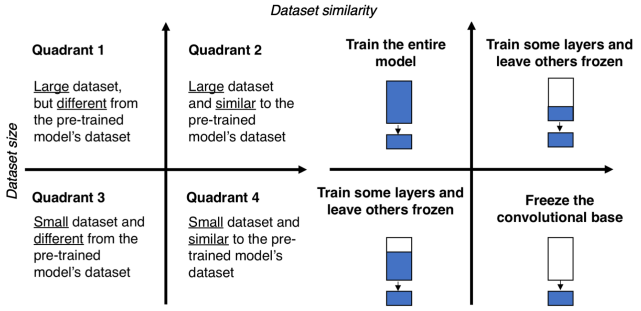

Figure 4. Size-Similarity matrix and fine-tuning [6]

Two different architectures are compared - VGG16, a CNN with 16 trainable layers of with 13 are convolutional layers, and ResNet50, a 50 layer CNN that has 47 convolutional layers. Both models consist of two parts: convolutional base and classifier with a number of fully connected (dense) layers. The main difference lies in the number and structure of convolutional layers.

Since the input shape of emotion images (48x48x1) varies from the input of pre-built models (usually, 224x224x3), the dense layers of them cannot be imported for the emotion recognition problem. For that reason, the architecture of the dense part is manually added upon convolutional layers of the pre-built models. Besides, grey-scale images were transformed into three-channel in order to adapt to the pre-trained models.

The VGG16 and ResNet50 were originally used for the prediction of 1,000 classes of "ImageNet" data, so for this problem, the architecture was adjusted to classify seven emotions by changing also the final dense (softmax activated) layer.

3.3.1 Denoising Autoencoders

Autoencoders are a special form of a neural network that contain an encoder to learn a latent representation of the input and a decoder to reconstruct the input from the latent space. The latent space is often a vector that is significantly smaller than the input and output forcing the non linear model to learn the most representative features of the input and compress them into the latent vector. This vector then holds enough information for a decoder to learn how to reconstruct the original input. Training loss selection is computed depending on the desired output, but in the case of normalized grey-scale images, binary cross entropy should be used after passing the output through a sigmoid function.

Autoencoders have many applications, but for the task at hand, the denoising autoencoders have a particular use case. The denoising autoencoders will take as input a noisy image to learn the latent space, but calculate the loss of the reconstructed image on a non-noisy version of the image. This allows the parameters of the model to learn how to remove noise from images. [**?**]

3.3.2 Other Denoising Methods

Convolutional Neural Network models learn the necessary convolutions to be able to classify images. If the models are trained on noisy data, they are capable of learning the appropriate convolutions to classify the right image despite noise. [7]

Median convolution filters take the most central pixel value within the kernel (3x3 in this case) and when sliding along each pixel of the image, the black or white noise resent in the image will be removed as they will likely never be the median values of the convolution. This form of filter will remove small specks of noise, but not be able to reconstruct blocks of noise. [4]

Lastly, Block-matching and 3D filtering (BM3D) is a denoising method that groups similar fragments of the image through a methods call block-matching. All image fragments are stacked into 3d cylinder shapes then are filtered and reconstructed into back to 2d by taking weighting overlapping images. [3]

3.3.3 Self-Training

Self-training is a technique of semi-supervised learning where a small set of labelled data is used to make predictions on a larger unlabeled data set to automatically generate labels. This approach eliminates the need for an expert to hand label the data which is slow and expensive. The newly labeled data is combined with the original labeled data and the model is re-trained. This general framework can take on

many forms such as Generative Adversarial Networks that leverage concepts of game theory or Contrastive Learning.

The approach for this report differs from a typical self-training implementation given there is already a well trained and working model that only needs fine tuning to real world images as they are collected. The output of the model on new unlabelled real world images will be a softmax probability for each emotion. To ensure only highly confident automated labels are kept for future fine tuning, a threshold is applied to the maximum classification probability per image. This threshold is a hyper parameter that needs to be determined through experimentation and heuristics. [8] Online learning methods were considered but not implemented given the approaches being explored are in the early stages and the experiments are using static data at this time.

3.3.4 Face Detection

One of the first real-time object detection system was a Harr cascade classifier which can be applied to detecting faces. This is a pre-trained model that detects Harr features, narrows the features using Adaboost and leverages Cascade of Classifiers grouping to classify faces if all features pass the processing checks. [9]

A more recent pre-trained Caffe model is available in OpenCV. This model uses Single Shot-Mutibox Detector and ResNet-10. The benefits to this newer model should be faster to compute, better with occlusions, and better with side faces. [1]

3.3.5 Measuring Performance

As an accuracy metric for comparison of the models the mean Average Precision (mAP) is used that computes AP scores over each of classes and takes mean of it. It is the most applicable evaluation metric for classification tasks of unbalanced data because it ensures all classes are given equal importance. For this project there are target emotions (surprised, happy and scared) that are more important to detect.

The confusion matrix is used to evaluate the quality of prediction for each class. On the diagonal of the matrix lie cases for whose the predicted emotion is equal to the real, while off-diagonals show the number of mislabeled faces. The Recall (true positive rate) and Precision matrices are built based on the confusion matrix.

4. Experiments

As a first experiment the comparison of pre-trained models in terms of amount of frozen layers is conducted. In the second VGG16 and ResNet50 models are compared, and the best model among them is used for further implementation. After the selection of the best performing model, methods for denoising images of faces are evaluated to determine if the data should be passed through the denoising stage before training. Then a self-training test is conducted to identify the probability confidence threshold by which the labeling of data is regarded as most accurate to be used for automatic labeling of training data. Finally, two face detection algorithms are tested to determine the most accurate at extracting faces from the future CCTV footage.

4.1. Frozen and Unfrozen Models Comparison

In this section, two VGG16 models pre-trained on Imagenet are compared, one were all parameters are trainable and the other where only the last dense layers are trainable. Both models are compared against a base model which is a CNN with an architecture of 10 layers of which two are convolutional layers. The dense layers of all the three models have matching structure for equal comparison.

The results of this experiment can be seen in Figure 5. It shows the average precision of each emotion of all three models. As expected, the model with all trainable parameters performs the best in all categories with an mAP of 70% and a maximum average precision for happiness at 93% . The other emotions of interest fear and surprise reach an average precision score of 57% and 86% respectively. The second-best model is the base model with an mAP of 64%, which reaches average precision scores of 44% for fear, 87% for happiness, and 78% for surprise. The model with the frozen pre-trained parameters performs the worst with a maximum average precision of 60% for happiness and an overall mAP of 41%.

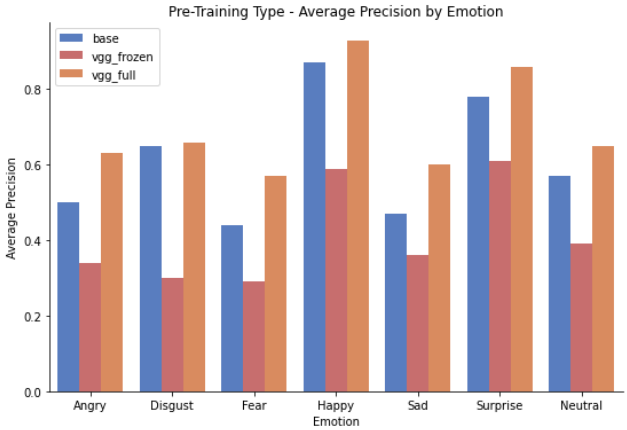

Figure 5. Pre-Training Results by Emotion

4.2. Pre-built CNN Comparison

As the previous section showed that the models benefit from trainable parameters, this section focuses on comparing two models with all trainable parameters, namely the VGG16 and ResNet50. The resulting average precision for the seven classes can be found in figure 6.

The comparison shows that VGG16 achieves the highest precision scores in all classes, an mAP of 70%. The emotions happy and surprise have very similar average precision across models (only 2-3 ppt lower for ResNet and 5-6 ppt lower for the base model). Fear has a lower precision value of all models ranging from 44% to 57% and shows greater discrepancies between the models. For all emotions apart from disgust, ResNet is performing better than the base model with an mAP of 64%. The base model reaches a slightly lower mAP of 61%. With respect to the running time, the base model is the most efficient, and ResNet the least efficient due to the complexity of the model architecture and the number of trainable models.

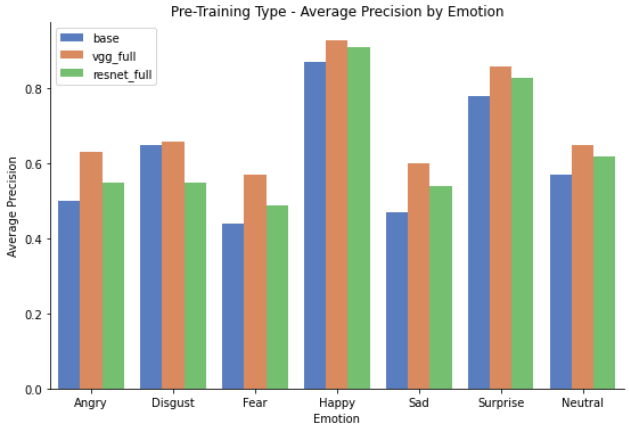

Figure 6. Architecture Results by Emotion

4.3. Image Denoising Performance

As mentioned in section 2.4, noise is applied to the data to experiment on the best method for denoising images of faces. Three methods are applied and measured using peak signal-to-noise ratio (PSNR) where a higher score indicates better reconstruction quality of an image. The following images are compared to the original image:

- No Reconstruction (baseline)
- Median Convolution
- BM3D
- Denoising Autoencoder (2-3x3 convolutions and 2-3x3 transpose convolutions)

The results in Figure 8 indicate that the autoencoder method has the best reconstruction with a PSNR of 75. The MSE of the autoencoder is 73% lower than the second best MSE (Median).

Proceeding with the denoising autoencoder, the next experiment compares the mAP prediction performance of the denoised images in the CNN model. The following methods are compared:

Figure 7. Image Denoising Results

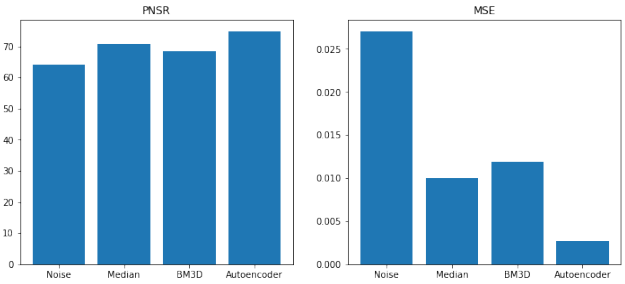

Figure 8. Denoising Loss Results

- Fine-tuned VGG model (Base)
- Non-noise images decoded using autoencoder then passed into base model (AE No Noise)
- Noise images passed into base model (Noise)
- Noise images decoded using autoencoder then passed into base model (AE Noise)
- Noise images passed into new fine-tuned VGG model retrained with noise images (AE Noise)

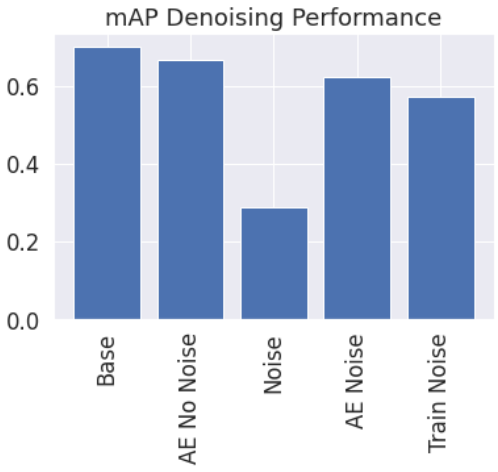

Figure 9. Image Denoising Model Performance

The results in Figure 9 show that the autoencoder does not hurt performance of non noisy image prediction (-5% reduction in mAP over baseline). The results also indicate that the autoencoder significantly increases the mAP of noise images if passed through the autoencoder first (117%

increase in mAP compared to Noise images). Lastly, the CNN trained on the noise images does show improved performance over the Noise model, but it has an mAP -8% lower than the model that first passes the noisy images through the autoencoder. In the end, it appears most effective to pass data through the denoising autoencoder before making predictions.

4.4. Self-Training Confidence Threshold

Given real facial expressions in the attractions will likely differ from the lab based emotions, fine tuning on new data will help improve the accuracy of the CNN model. Also, the distribution of surprised emotion classified images is lower than desired given the task and more data needs to be added over time. Unfortunately, in order to fine tune on real data a massive effort needs to be made to label data. Instead of manually labeling data, this experiment will leverage the existing fine tuned VGG model to classify newly detected faces. If they pass a threshold of probability, they will be deemed as confident and correctly labeled by the model. The new labeled data that pass threshold will be added to the input data for future training. The experiment will help decide the ideal threshold by comparing the number of images and mAP by each 10% probability threshold from 0% to 90%.

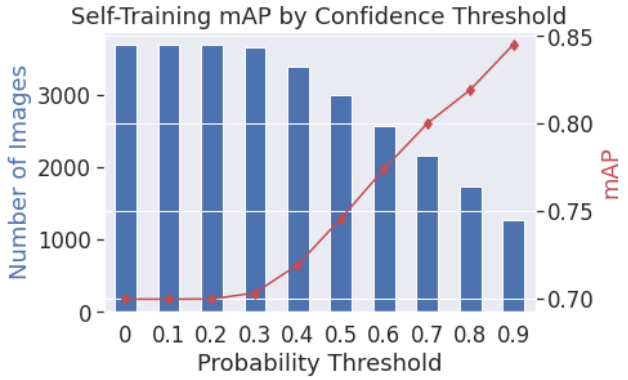

Figure 10. Self Training by probability threshold

The results in Figure 10 indicate that the highest mean average precision occurs at a probability threshold for of 0.9. At this threshold, 34% of new faces will be retained for future training and with less than 16% false classifications.

4.5. Face Detection

The footage captured will have multiple faces in each image that all need to be detected. Two pre-trained face detection algorithms are tested (Harr and DNN Caffe) to determine which one will be the best fit for the given task. For this experiment, the face detection is applied to the scrapped Bing images and manually checked for accuracy. In total, there are 244 possible faces to detect from the 95 images. The Harr model detects 66% of the possible face while DNN Caffe detects 68%. Despite their coverage being similar, the DNN Caffe model has 0 faces incorrectly detected while 8% of the Harr detections are not faces. In addition, Figure 12 showcases that the DNN Caffe model is better at placing boxes around the entire face than HARR (Figure 11). Overall the DNN Caffe model is the stronger performer.

Figure 11. HARR Cascade face detection

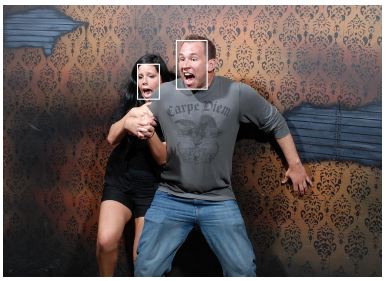

Figure 12. DNN Caffe face detection

5. Conclusion

Piecing all of these experiments together, the fine-tuned VGG16 model with manually added dense layers achieves the best results as the starting model for detecting emotions in a haunted house. Among all tried models it gives the best results in terms of predicting classes which are most important for the task.

Faces will be detected from the CCTV security footage using the preferred DNN Caffe model. Once detected the face will first be passed through the denoising autoencoder to ensure any noise is reduced before then being passed into the CNN model to predict the emotion. The objective is to maximize the number of surprised or fear emotions and minimize angry and neutral reactions. Any faces that have a maximum probability greater than 0.9 will then be accepted as newly labeled data that will be added to the input images used for the next batch of model training.

References

[1] Vardan Agarwal. Face detection models: Which to use and why?, Jul 2020.

[2] Pierre-Luc Carrier and Aaron Courville. Challenges in representation learning: Facial expression recognition challenge, 2013.

[3] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[4] Robert Fisher, Simon Perkins, Ashley Walker, and Erik Wolfart. Median filter, 2003.

[5] Patrick Lucey, Jeffrey F. Cohn, Takeo Kanade, Jason Saragih, Zara Ambadar, and Iain Matthews. The extended cohn-kanade dataset (ck): A complete dataset for action unit and emotion-specified expression. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, 2010.

[6] Pedro Marcelino. Transfer learning from pre-trained models. *Towards Data Science*, 2018.

[7] SANDRO LOCKWALL RHODIN and ERIC KVIST. A comparative study between mlp and cnn for noise reduction on images. *KTH ROYAL INSTITUTE OF TECHNOLOGY SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE*, Jun 2019.

[8] Doug Steen. A gentle introduction to self-training and semi-supervised learning, Aug 2020.

[9] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*.