An Exploration on the Applicability of a Pilot-Scale Facial Recognition Model in Live Video

Bojing Yao
Toronto Metropolitan University
CP8307
Toronto, Canada
bojing.yao@torontomu.ca

Abstract— **In recent years, Machine Learning and, specifically, Convolutional Neural Networks (CNN) have become instrumental in image processing tasks due to their efficiency and accuracy. This study delves into the application of CNNs for real-time facial recognition and labelling in videos. Utilizing a deep learning architecture, we applied Transfer Learning to a pretrained model, enhancing its performance for small scale, light-weight applications. We conducted a pilot project to ascertain the model's efficacy, achieving an accuracy of over 85% against a testing dataset and successfully labelling faces with over 90% accuracy in a controlled video environment. While promising, a noted limitation is the processing time, which extends the original video duration by 17 times, underscoring an area for future optimization. Facial recognition has always been an area with a lot of applicability, whether be it in school to detect student's behaviours, or at work, analyzing employee's performances. This report aims to evaluate the performance of a pilot scale facial recognition model with VGG16 Convolutional Neural Network (CNN) for a light-weight usage in real time videos.**

The code used for this project is hosted at this address: https://drive.google.com/drive/folders/1v1DBIXixAvDO0RQ-e1c_ORzL6Nwv5Xqf?usp=sharing

Keywords—Convolutional Neural Networks, Machine Learning, Image Processing, Deep Learning, Transfer Learning

I. INTRODUCTION

Facial recognition has always been an area with a lot of applicability, and recent advances in technology has allowed these technology to become reality. [4] The challenges of facial recognition remains mainly in the amount of data available and the ability for the model to process and correctly differentiate between different labels each with few data, while the data themselves are similar to each other and may be in non-optimal qualities. Obtaining and processing the vast amount of data is a difficult task which is usually done by companies with a lot of resources to spend, however, smaller organizations could also benefit from this technology, therefore another main challenge would be the cost of implementing this novel technology in a small and reliable format that is economically and structurally acceptable for smaller organizations to utilize. With rapid research and innovations being produced that allows a more streamlined and standardized machine learning workflow, [5] it is very realistic to apply CNN technology on a small-scale. This technology's application would mainly for security and education, as it proves the potential to create a more secure environment, and it is a better tool for individual's records keeping.

There have been a number of work already done on topics related to this project. In 2014, a research done by Yaniv Taigman et al. has utilized a CNN model with more than 120 million parameters for feature extraction featuring a novel 3D alignment approach that supports the model's flexibility in analyzing various poses. [1] The model was trained on over 4 million facial images of more than 4000 person. This model achieved a 97% accuracy on labelling faces with the Labeled Faces in the Wild (LFW) dataset. This approach involves facial landmark detection to obtain a unique overview of subjects' faces thus reduced intra-personal variations. In 2015, Florian Schroll et al. presented a new method using a CNN with a triplet loss function to optimize the embedding space for discriminability. [2] This research aimed to distinguish identities better. The model featured in this research was also trained on a very large dataset. To finish it off, Jason Yosinski et al. has produced a research paper that investigated the ability for Deep Neural Networks to transfer the already trained knowledge to be applied on new layers.[3] The result of this research has indicated that features learned by the Deep Neural Networks are highly transferable to newer layers, being affected negatively by the specialization of higher layers.

It has been established that CNN models are capable of generating excellent results in facial recognition and labelling tasks while offering a great amount of flexibility. [6] Since project as defined in this paper aims to produce a solution which achieves high accuracy in labelling faces and is lightweight in terms of training parameters and data required, it is clear that a Deep Learning CNN architecture would be necessary. Amidst these advancements, a gap persists for an economical, yet efficient facial recognition model tailored for smaller organizations, balancing cost and performance. This project aims to bridge this gap, utilizing a combination of deep learning and transfer learning to achieve high accuracy while being economical in data and computational requirements.

This project is set to produce significant to popularize facial recognition technology. By utilising a pre-trained CNN model and applying transfer learning, this project offers a solution that is not only efficient in terms of accuracy, but also lightweight in terms of economically and technological considerations for a wide range of organizations. One of the primary contribution of this project is the adaptation of a complex technology into a

much more lightweight and accessible format. The model provides a solution in facial recognition without the need for extensive computation or an extensive dataset resources. Also, this project introduces a novel approach to real-time facial recognition by adding in a feature for emotion recognition. This integration of emotion and facial recognition adds a sophisticated layer for applications such as security surveillance and identify verification processes.

At it's core, this project aims to be a testament to the adaptability and versatility of CNN facial recognition technology. The balance of accuracy and usability shows a significant leap towards a universal access to this technology with much broader applications.

II. Technical Analysis

A. Background Theory

This technical analysis presents the theory of a Deep Learning CNN with Transfer Learning architecture. Neural Networks (NN) are a type of computational models inspired by the human brain's structure, which is composed of layers of interconnected neurons, with each neuron capable of processing input data and producing an output through an activation function. NN learns by adjusting the weights of each neurons, based on the error of the model's prediction in comparison to actual outcomes. These types of models are capable of modelling complex and non-linear relationships.

The CNN, on the other hand, is a type of NN with specialized convolutional layers. With the convolutional layer, the CNN is capable of feature extraction of complex images. A set of filters or kernels is applied to the input image through convolution, with each individual kernel or filter designed to detect specific features, such as edges or patterns. Typically, a 3x3 convolution matrix is used, where K is the kernel mask and I is the image: [8]

$$S(i, j) = (I * K)(i, j) = \Sigma_{m=-1} \Sigma_{n=-1} I(m, n)K(i - m, j - n).$$

The convolutional layers works by applying the kernel to the image several times to identify key attributes. After each layer, a feature map is determined, and passed through an activation function (ReLU) for evaluating weights, and between each convolution layers, the feature map is down sampled to produce a more information-dense input for the next layers. What affects the learning of the model is the weights of the different filters. The weights are initialized randomly and optimized during each epoch through backpropagation. This process refines the weights and allows the model to learn the features of the input data. The architecture of a CNN usually has the lower layers capturing edges and colors, while having higher layers capturing more complex and abstract features. The CNN models have been shown to perform well in identify abstract high level features, especially when coupled with a deep learning architecture, the identification of features evolve through layers and eventually adapt to the training set to identify key features in testing images. [7]

There is also the pooling layer. Essentially, the pooling layer serves as a reduction method to reduce the input dimensions. This allows for a decreased computational complexity of the model, and allowing feature detection to be independent of the scale.

Deep Learning is often used in CNN, as it allows for a feature processing in a hierarchical way. In this project, Deep Learning is especially useful as it allows utilization of previously trained model and enables the addition of more layers that are tailored to our specific use case. Deep Learning models consist of multiple hidden layers that are designed to recognize different features from the input data. The depth of the NN supports the model in capturing abstract patterns and more complex features from the data. With facial recognition, this architecture allows the model the distinguish subtle facial features and expressions. With our model architecture, initial layers are from a pre-trained model which already has low-level features such as edges and colours learnt, we can transfer the available knowledge from the lower layers to our added layer, which is specialized on facial recognition tasks with our given dataset.

B. Experimental Methods

In this project, the VGG16 model is chosen as the base model to be expanded upon. The VGG16 model, originally developed by the Visual Graphics Group, is known for its simplicity and efficiency. The model is available with pre-trained weights with the ImageNet dataset, which is a large dataset of more than 14 million images across 1000 categories. The VGG model uses 3x3 convolutional filters for feature identification at each layer. This allows transfer learning to be applied in our new layers, saving valuable resources. VGG16 already has multiple built-in convolutional layers, thus any newly added layers can be used to learn more high level and abstract features. [6]

Moreover, a systematic approach was developed to preprocess, train, and validate a model based on our use case. The training process is started by creating an acceptable folder structure. In our project, the training images are images of celebrity's faces obtained from online sources, in various lighting conditions. There are around 100 images for each of the celebrity used for this project. The images are under different lighting conditions, in different environments and with different clothing.

Fig. 1. Example training images used in the data

The variety of conditions in the images tests the versatility of the model and allows the models to identify key facial features in the classification task, instead of the clothing styles or other accessories. The data are put into folders with their corresponding names, the script iterates through those folders to process the images into a dimension of 224x244 pixels. The images

```
[[1. 0. 0.]
 [1. 0. 0.]
 [1. 0. 0.]
 ...
 [0. 0. 1.]
 [0. 0. 1.]
 [0. 0. 1.]]
```

are also converted arrays, and preprocessed to ensure their shape are consistent with the training requirement. A list of labels is also generated in the format of one-hot encoded array, as shown in the figure on the right.

Then, data (already converted to arrays), is split three ways, into training data, testing data, and validation data, thus concludes the preparation phase.

All layers in the base model is locked as the idea is to add onto the model instead of retrain the model. Additional dense layers are appended, and since the goal is to use the features already extracted for a classification task, a flatten layer is first used to convert the inputs to a 1D array. The model is trained on the dataset for 20 epochs, utilizing the Adam optimizer and categorical cross-entropy loss function, which is suitable of the classification task. Throughout the training epochs, the model is validated on the validation set, ensuring accuracy and avoiding overfitting, also to provide insights on the model's learning rate. The training phase is concluded with testing the model against a curated dataset, and using graphs to evaluate the model's learning capabilities, and accuracy to measure the model's performance, details will be provided in the Experiments section of this report.

The model's applicability is evaluated for it's integration into a video processing script. A cropped video of one individual's face is fed into the script, with each frame of the video extracted, preprocessed, and fed into the model for making real-time predictions. This work flow also utilizes the DeepFace library's emotion recognition package, in order to produce a meaningful result. After which, the process' metrics are computed and analyzed into assessing this project's capabilities in a real-world scenario.

III. Experiments

A. Baseline Model Training

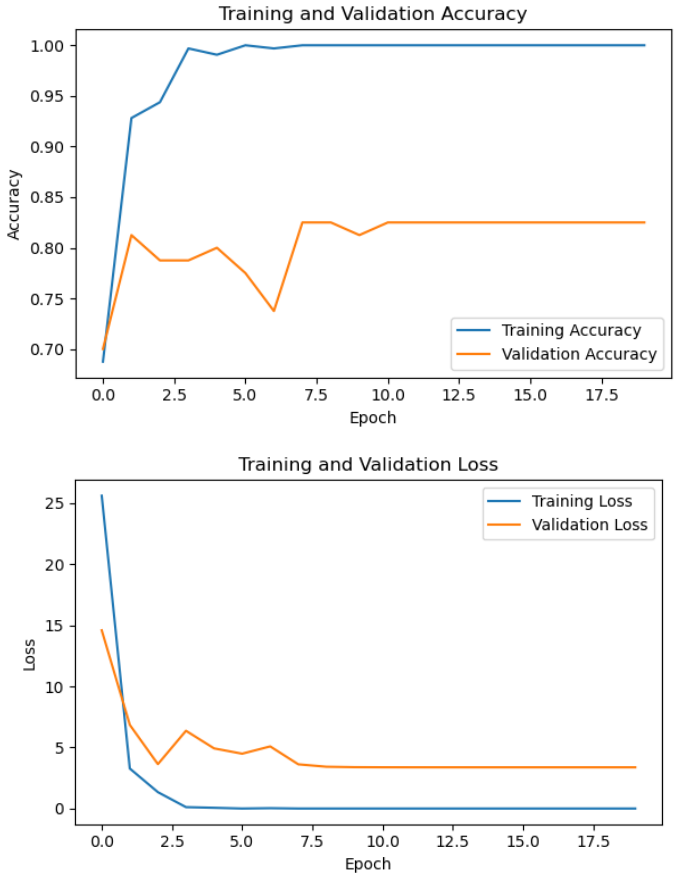
During the experiment stage of this project, a systemic approach was adopted to preprocess, train and validate the model with a dataset of images, utilizing the VGG16 architecture combined with one flatten layer to convert the image to a 1D array, one dense layer to process the parameters from previous layers, and one labelling layer, to process the weights and label the image based on the faces learned. The process is first started with the iteration over each sub-directory, adding the image to a list while also adding the corresponding hot-encoded labels to another list. The image is preprocessed to ensure the shape is consistent, and the pixel dimensions are converted to be 224 by 224 pixels. The baseline model uses 80 training images for each classification category, 240 images in total across three different celebrities.

TABLE I. Training Image Codes

Name	Coded Array for Image Labels
	Code
RDJ	[1. 0. 0.]
Scarlett	[0. 1. 0.]
Tom	[0. 0. 1.]

Then, the images are split into training and testing sets with a 0.2 test split ratio, to ensure maximum number of samples are used for training. The base model uses pretrained weights from 'imagenet'. A new sequential model is created to encapsulate the VGG16 base model and our additional dense layers. This approach aims to apply transfer learning, using the original feature mappings learned by VGG16 from the imagenet dataset on the new samples. Using the Adam optimizer and categorical cross-entropy loss function, the model is trained on the dataset for 20 epochs. During the training, the model is validated on a separate testing dataset, which provides insights into the model's capabilities.

Fig. 2. Training and validation accuravy vs Training and validation loss



As seen by the diagram above, the model achieves a very consistent validation accuracy after 10 epochs, and the validation loss achieves a minimum after 8 epochs. This data suggests that 10 epochs are sufficient for training this model for our dataset. Training accuracy reaches 1.0 at epoch 7, which validation reaches 0.825 at epoch 9 and decreases at epoch 10. The accuracies are maintained at these values from epoch 10 to 20. This may suggest that the model is overfitting a little bit considering the difference in training accuracy and validation accuracy, however the validation accuracy is acceptable for a

few shot model with a simple architecture and a small sample size. Throughout the epoch, the training loss decrease at a rate that is as expected, suggesting the learning rate and loss function used is suitable for this scenario.

TABLE II. EPOCH VS ACCURACY COMPARISON

Epoch	Epoch Training/Validation Accuracy		
	Training Loss	***Train Accuracy***	***Val Accuracy***
1	25.6154	0.6875	0.7
2	3.2748	0.9281	0.8125
3	1.3407	0.9438	0.7875
4	0.1134	0.9969	0.7875
5	0.0603	0.9906	0.8000
6	1.9402e-06	1.0000	0.7750
7	0.0277	0.9969	0.7375
8	1.8552e-07	1.0000	0.8250
9	5.2154e-09	1.0000	0.8250
10	7.4506e-10	1.0000	0.8125

Based on the result obtained above, any improvement to the model can be experimented on changing the model architecture to add in more layers to learn more abstract features on the dataset, or to unfreeze some layers from the pre-trained VGG16 model to let the model's higher layers learn more about the sample data. The model's capabilities is further explored through a hand picked testing dataset that is unseen to the model. The result indicates the model performs very well against an unseen dataset. However, it is possible for this performance be due to the hand picked dataset being too similar to the training set, since the pictures of celebrities obtained from online sources are very similar in terms of the image quality and layout. The resulting model achieves a 93% accuracy against testing data that is unseen to the model.

TABLE III. TESTING ACCURACY BASELINE

Name	Epoch Training/Validation Accuracy			
	Precision	***Recall***	***F1-score***	***Support***
rdj	1.00	0.75	0.86	4
scarlett	1.00	1.00	1.00	4
tom	0.80	1.00	0.89	4
Accuracy			0.92	12
Macro avg	0.93	0.92	0.92	12
Weighted avg	0.93	0.92	0.92	12

B. Reduced Training Samples Evaluation

Since the objective of this project is to explore a lightweight strategy of training, a significantly less training data is also used to train the model, in order to evaluate the versatility of the model. In this section, a combined training data for three celebrity categories of 150 images and a training data of 75 images are used and the model's performance is evaluated.

TABLE IV. TESTING ACCURACY 30% DATA REDUCTION

Name	Epoch Training/Validation Accuracy			
	Precision	***Recall***	***F1-score***	***Support***
rdj	0.80	1.00	0.89	4
scarlett	1.00	1.00	1.00	4
tom	1.00	0.75	0.86	4
Accuracy			0.92	12
Macro avg	0.93	0.92	0.92	12
Weighted avg	0.93	0.92	0.92	12

TABLE V. TESTING ACCURACY 60% DATA REDUCTION

Name	Epoch Training/Validation Accuracy			
	Precision	***Recall***	***F1-score***	***Support***
rdj	1.00	0.25	0.40	4
scarlett	0.67	1.00	0.80	4
tom	0.40	0.5	0.44	4
Accuracy			0.58	12
Macro avg	0.69	0.58	0.55	12
Weighted avg	0.69	0.58	0.55	12

As seen by the result in table IV and table V, after a 60% reduction (75 training data, 25 images for each person) in the size of the training data, the model is only performing slightly better than guessing, suggesting that the model is unable to learn effectively based on the number of samples and also on the quality of the samples. A 30% reduction in training data (150 training data, 50 images for each person) is still in the acceptable range for the model to perform on-par with the baseline model.

C. Live Video Application

For obtaining the best possible results, the following section uses the baseline model. For using the model in real life applications, this project has obtained a video from the internet for the model to be test against. The video is cropped from the original such that only the person's face is present in the frame. Every frame of the video is preprocessed and evaluated using the model, and the model responds with either the correct name of the person, ex: "Tom", or with "Unknown". Another package that can indicate a person's emotion is also used during

For using the model in real life applications, this project has obtained a video from the internet for the model to be test against. The video is cropped from the original such that only the person's face is present in the frame. Every frame of the video is preprocessed and evaluated using the model, and the model responds with either the correct name of the person, ex: "Tom", or with "Unknown". Another package that can indicate a person's emotion is also used during this part of the project to explore the model's applicability when used with external tools.

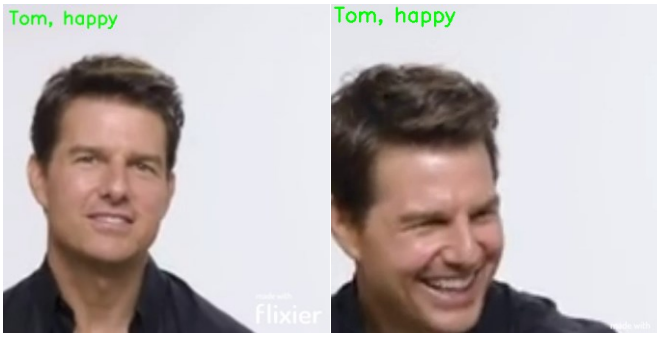


Fig. 3. Example video screenshot, with correct emotions labels and name labels

The model is able to achieve an accuracy of 78.42%, which means the model is able to recognize the correct face in the video at around 78% percent of times, when processing every frame of the video. This result achieved is slightly lower than the validation accuracy, however, this is still accurate enough to be used in identifying a person from a video given enough optimization to the code. A significant issue encountered by this model is the speed at which it runs. The original length of the video is 6.3 seconds long, and the total time it takes to process every frame of the video and reconstruct it with labelled faces takes 106.59 seconds, which is around 17 times the original length of the video. Running the model on each frame of the video also reduce the frame rate from 30 frames per second to 1.79 frames per second. This shows area that can use significant improvements, possible optimizations include evaluating the video on random steps and running the model as an asynchronous process, instead of bottlenecking the displaying of the video.

IV. Conclusion

In this project, we explored and implemented a facial recognition model mainly based on VGG16. We have tested it's versatility by tailoring the model with our own limited samples, and using it to solve specific classification task. Through the process, we have generated a refined model capable of correctly classifying faces with a 78% against targets in a video. We have also identified that the model architecture produced is very versatile in terms of working with limited training data. The optimal range of training data that will allow the model to perform well lies between (40-50) images per person. The model shows promising results to be used on a small dataset with limited variety. The combination of the pre-trained VGG16 model, transfer learning techniques and the addition to the architecture produced a model that is effective for our use case.

There are many possible applications of this refined model technique. In security and surveillance sectors, the technique shows promising facial recognition capabilities, industries can use this technique to construct a more versatile model that is specifically catered to their employees, while extending the project's functions, in order to better understand their employees' daily routines and to better monitor their employees' mental conditions through emotion detection. Moreover, in healthcare industries, this technique can be used for patient monitoring and emotion analysis, suitable for non-verbal patients or for detecting underlying emotions. This technique can also be augmented with any facial recognition functionalities, further extending it's use.

Recommendations for future work based on this project includes optimizing the algorithm when coupling the model with a live video, since analyzing the video's every frame does not yield a good efficiency. The latency can be reduced by limiting the samples during a live video, and converting the model to be ran on asynchronous formats. Furthermore, the model's architecture should be explored deeper to fine tune the accuracy in correct facial recognition, and datasets with greater variety should be tested with this model to improve it's accuracy in facial recognition under all lighting conditions and poses.

References

[1] Y. Taigman, M. Yang, M. Ranzato and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 2014, pp. 1701-1708, doi: 10.1109/CVPR.2014.220.

[2] F. Schroff, D. Kalenichenko and J. Philbin, "FaceNet: A unified embedding for face recognition and clustering," 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 2015, pp. 815-823, doi: 10.1109/CVPR.2015.7298682.

[3] Yosinski J, Clune J, Benigo Y, and Lipson H. *How transferable are features in deep neural networks?* In Advances in Neural Information Processing Systems 27 (NIPS' 14), NIPS Foundation, 2014.

[4] Parkhi, O. M., Vedaldi, A., Zisserman, A., & Jawahar, C. (2015). Deep face recognition. In British Machine Vision Conference

[5] Abadi, M., "TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems", arXiv, 2016. doi:10.48550/arXiv.1603.04467.

[6] A. S. Razavian, H. Azizpour, J. Sullivan and S. Carlsson, "CNN Features Off-the-Shelf: An Astounding Baseline for Recognition," 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops, Columbus, OH, USA, 2014, pp. 512-519, doi: 10.1109/CVPRW.2014.131

[7] Zeiler, Matthew D. and Rob Fergus. "Visualizing and Understanding Convolutional Networks." ArXiv abs/1311.2901 (2013): n. pag.

[8] I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning. MIT Press, 2016, ch. 9. [Online].