



AN IMAGE CLASSIFIER FOR
HEADSHOT PHOTOGRAPHERS

Introduction

WHY?

- ▶ I have a side business as a photographer, focusing on headshot photography for film and theatre actors.
- ▶ Headshot photographers will typically take 300-600 photographs in every session.
- ▶ Invariably, the subject is blinking, or has partially closed eyes in some of those photographs.
- ▶ After every session, I manually edit out these photographs before sending them to the client, which can take 30 minutes to an hour for each session.
- ▶ I was considering hiring an assistant, but I wanted to see if I could create a deep learning algorithm to save me time and expense.

Introduction

OBJECTIVE

- ▶ To build a convolutional neural network that correctly classifies images where the subject is blinking with 80% accuracy.

PROJECT STEPS



.01

Data Collection

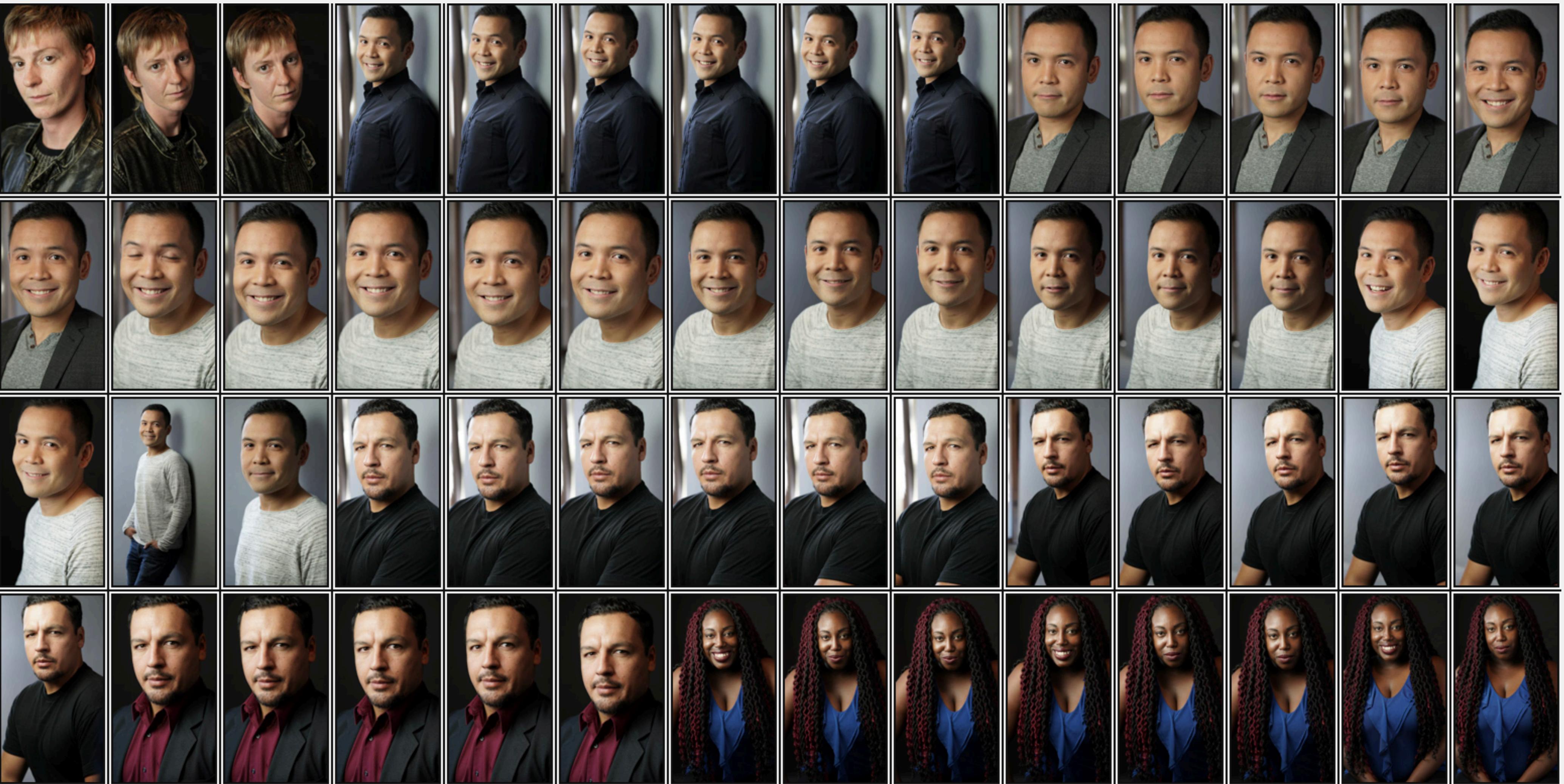
SOURCE

- ▶ I manually labelled photographs from my own collection of archived sessions, splitting them into BLINKING and NON_BLINKING groups.
- ▶ Each group included 500 images.

SUBJECTS

- ▶ I aimed to create a dataset that was diverse in age, gender, and racial groups. The approximate breakdown was:
- ▶ **Female: 54%, Male 43%, Non-Binary: 3%**
- ▶ **White: 60%, Black: 27%, Asian: 8%, Other: 3%**
- ▶ **Age < 25: 8%, 25-50: 49%, 50+: 38%**

NON-BLINKING



BLINKING



Model Creation & Tuning

.02

IMPORTING DATA

- ▶ I pre-scaled the images to save processing time and expense.
- ▶ For the privacy of my clients I am not including the images on github, but this is an example of 'blinking' and 'non-blinking' photos from the data set:



Model Creation & Tuning

.02

INITIAL MODEL

Train Accuracy: 0.817, Test Accuracy: 0.777
Precision: 0.778523
Recall: 0.773333
F1 score: 0.775920

Model: "sequential_6"		
Layer (type)	Output Shape	Param #
conv2d_12 (Conv2D)	(None, 176, 98, 256)	2560
activation_18 (Activation)	(None, 176, 98, 256)	0
max_pooling2d_12 (MaxPooling)	(None, 88, 49, 256)	0
conv2d_13 (Conv2D)	(None, 86, 47, 256)	590080
activation_19 (Activation)	(None, 86, 47, 256)	0
max_pooling2d_13 (MaxPooling)	(None, 43, 23, 256)	0
flatten_6 (Flatten)	(None, 253184)	0
dense_12 (Dense)	(None, 64)	16203840
dense_13 (Dense)	(None, 1)	65
activation_20 (Activation)	(None, 1)	0
<hr/>		
Total params: 16,796,545		
Trainable params: 16,796,545		
Non-trainable params: 0		

Model Summary

Model Creation & Tuning

.02

TUNING

- ▶ Tuned using Keras Hyperband tuner

The hyperparameter search is complete. The optimal number of units in the first densely-connected layer is 256 and the optimal learning rate for the optimizer is 0.0001

```
model.summary()  
  
Model: "sequential"  
  
Layer (type)          Output Shape       Param #  
=====              ======           ======
```

flatten (Flatten)	(None, 17800)	0
dense (Dense)	(None, 256)	4557056
dense_1 (Dense)	(None, 10)	2570

```
Total params: 4,559,626  
Trainable params: 4,559,626  
Non-trainable params: 0
```

Final Model

.02

Model Creation & Tuning

FINAL MODEL

Train Accuracy: 0.86, Test Accuracy: 0.79

Precision: 0.77

Recall: 0.82

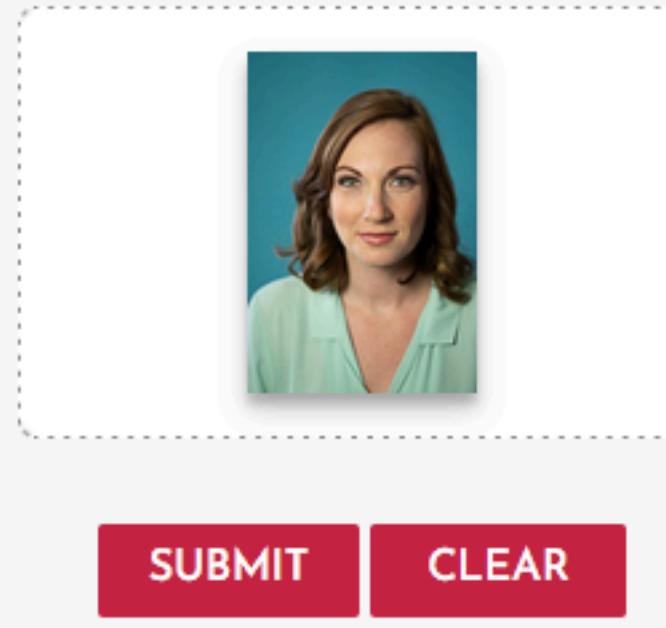
F1 score: 0.775

App Deployment

.04

FLASK APP, DEPLOYED TO HEROKU

EYE SPY: An Image Classifier for Headshot Photographers



MVP version of the app

Conclusion

- ▶ The convolutional neural network was able to classify blinks versus non-blinks with 79% accuracy on the testing data set.

CAVEATS

- ▶ My data set included 500 blinking examples and 500 non-blinking examples. I believe I could achieve a higher accuracy rate with a larger data set.

Conclusion

NEXT STEPS

- ▶ Go back to the initial model, and see which images are being misclassified. I can then provide more training examples of similar images, which may improve model performance.
- ▶ I'd like to build a more robust app that will import a large number of images, and filter them into two groups for final review.
- ▶ This could either be a web app, an MacOS app, or a plugin to an existing photo editing application.

CONTACT

lana@lanapalmer.com

lanapalmer.io