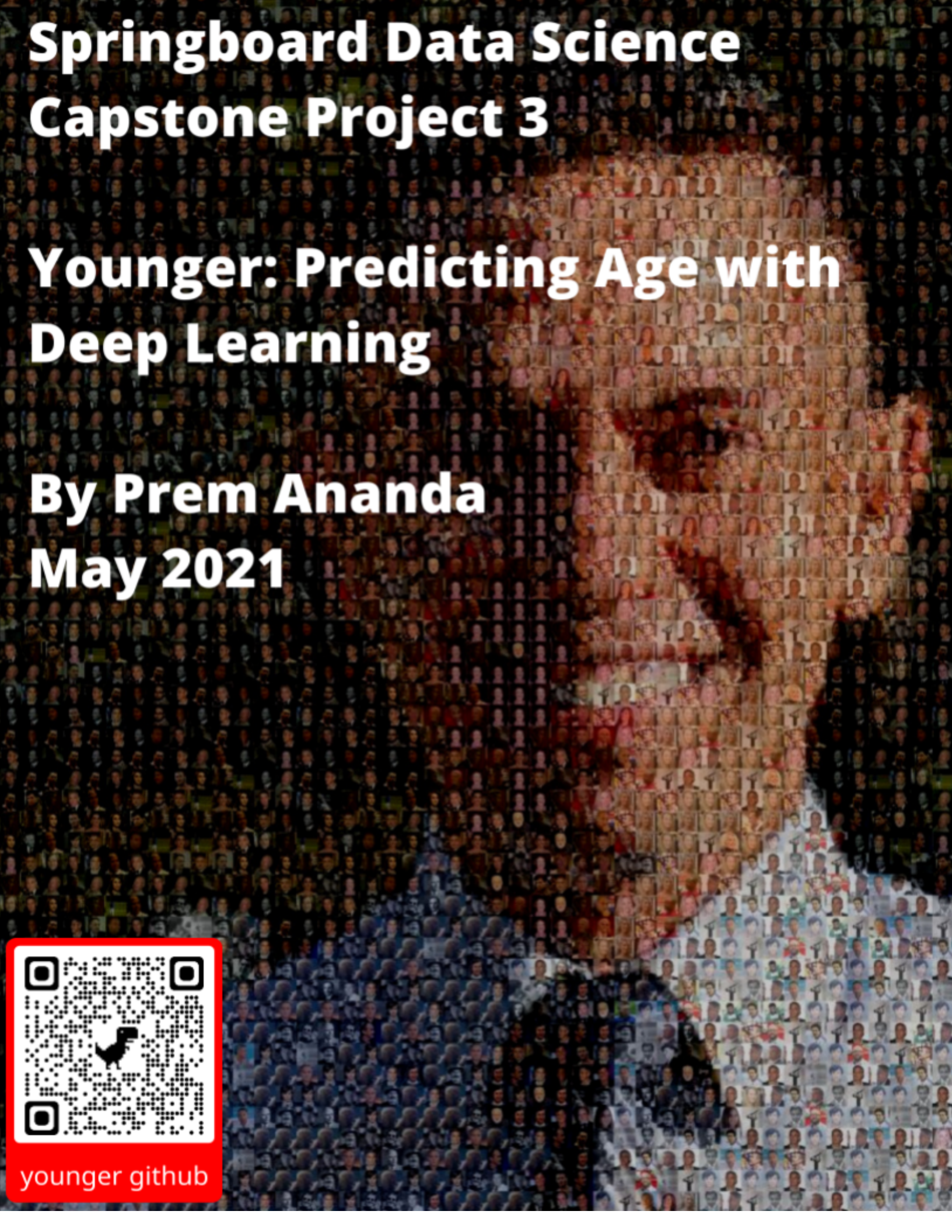
Springboard Data Science Capstone Project 3

Younger: Predicting Age with Deep Learning

By Prem Ananda

May 2021



younger github

Younger: Predicting Age with Deep Learning

Executive Summary

Problem: The problem is to develop and evaluate image-based supervised models to predict the age of the person in a given image, using deep neural nets.

Hypothetical Stakeholder: A beauty company, 'Younger', requested an age predictor app to demonstrate the value of their 'age-defying' product line.

Image 1. Average of 1000 Female Faces.

Results: Using a customized Convolutional Neural Network (CNN) and Transfer Learning, we were able to estimate a subject's age from a photograph with an average error of about 7 years.

Recommendations: We can move forward by creating a Beta app with the disclaimer that the system is still being optimized. We recommend collecting more representative data, and spending more time fine-tuning the model. The system can be improved significantly with more high-quality, labeled data and more tuning. We expect that we can reduce average prediction error to about 5 years.

The Code

All project code is available on my Github repository.[1]
The project was developed in Python using appropriate libraries on Google Colab Pro[2] to take advantage of high-performance GPU-based architecture. TensorFlow and Keras were used for preprocessing and implementing the Convolutional Neural Network (CNN).

The Dataset

The dataset "IMDB-WIKI – 500k+ Face Images With Age and Gender Labels" related to this paper is available for download for academic research only.[3]

---

[1] Ananda, Prem. YOUNGER. Retrieved on May 18, 2021, from https://github.com/premonish/YOUNGER.
[2] https://colab.research.google.com
[3] Rothe, R., Timofte R., Van Gool, L. (2015). *IMDB-WIKI – 500k+ Face Images with Age and Gender Labels.* Retrieved on May 8, 2021, from https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/.

1. INTRODUCTION

Why Age Prediction?

Why would anyone want to predict people's ages automatically? Great question! We live in a world of AI face recognition. More and more we are seeing facial recognition being used for identification in various settings. In China, millions of people are paying with their faces daily.[4] Privacy concerns aside, we are interested in how this facial matching technology works. Accordingly, we will focus on one aspect of creating a facial profile: the age. A person's age can not be precisely estimated by other people. We could only make an approximation and categorize someone as "about 25" or "she looks about 50 years old". The challenge here is to take a large dataset of faces and use current technology to extract patterns. These extracted patterns can be used to automatically predict the age of different faces with reasonable accuracy.

Open Questions

1. Can we use public datasets and open-source software to create an age predictor with reasonable accuracy?
2. What are the major limitations and challenges when attempting to estimate age?
3. Are there any systemic biases in public datasets? How do we overcome any biases?

Current Uses of Age Prediction

Age predictions are used in marketing campaigns to play targeted ads. Quividi, for instance, has signage that senses a passerby's age and gender and plays an appropriate video advertisement.[5] In a light-hearted mobile app called "Check My Age", users input a photo and have their age predicted with the option of sharing the results on their social media accounts. According to the academic paper "DEX: Deep Expectation of Real and Apparent Age from a Single Image without Facial Landmarks" (hereafter 'DEX'), which helped inspire this project, "*Applications where age estimation can play an important role include: (i) access control, e.g., restricting the access of minors to sensible products like alcohol from vending machines or to events with adult content; (ii) human-computer interaction (HCI), e.g., by a smart agent estimating the age of a nearby person or an advertisement board adapting its offer for young, adult, or elderly people, accordingly; (iii) law enforcement, e.g., automatic scanning of video records for suspects with an age estimation can help during investigations; (iv) surveillance, e.g., automatic*

[4] Takashi, K. and Hinata, Y. (2019). *Pay with your face: 100m Chinese switch from smartphones.* Nikkei Asia. Retrieved on May 8, 2021, from https://asia.nikkei.com/Business/China-tech/Pay-with-your-face-100m-Chinese-switch-from-smartphones.

[5] https://www.kdnuggets.com/2019/04/predict-age-gender-using-convolutional-neural-network-opencv.html

detection of unattended children at unusual hours and places; (v) perceived age, e.g., there is a large interest of the general public in the perceived age (c.f. howhot.io), also relevant when assessing plastic surgery, facial beauty product development, theater and movie role casting, or human resources help for public age specific role employment."[6]

Challenges

The task of estimating the age of people from pictures of their faces is complex since there are many variables affecting a person's facial age appearance in real life. Some variables in apparent age are: sun exposure, ethnicity, wrinkles, age spots, lifestyle (e.g. smoking), complexion, face alterations, makeup, etc. Also, in a facial photo, there are several factors affecting apparent age such as lighting, angle, filters, expression, camera quality, etc. as anyone who has taken a "selfie" can attest (see Image 2).

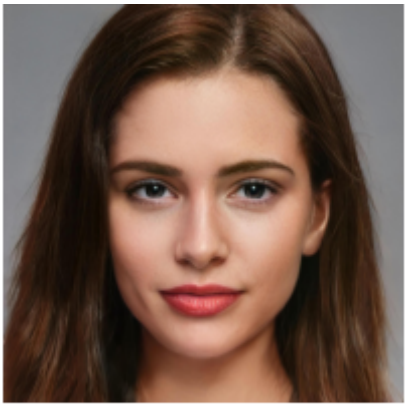


Image 2. *Guess her exact age.*
How confident are you in your estimate?
Photo by Art Hauntington on Unsplash

Facial Recognition in the USA

There is an ongoing debate in the USA about the privacy and ethical issues around facial recognition technology. A handful of the largest tech players have developed cutting-edge facial recognition software. Facebook has led the way when it introduced the feature of automatically suggesting 'tags' to friends' faces in a photo, which sparked privacy concerns. In 2020, Amazon decided to refuse the police the use of its facial recognition technology for at least 1 year."[7] IBM quit the facial-recognition business altogether. In a letter to Congress, IBM CEO Arvind Krishna condemned software that are used "for mass surveillance, racial profiling, violations of basic human rights and freedoms."[8] Google Photos uses extremely sophisticated facial recognition technology to help users identify, label, and search for their contacts in their personal image archives. Google Lens can quickly identify famous faces among thousands of other things. Microsoft has stated that they do not sell their facial recognition technology to law enforcement. Clearly, many large tech companies have developed facial recognition software and are concerned with the "correct" use of the technology.

[6] Rothe, R., Timofte, R., Van Gool, L. (2016). *Deep expectation of real and apparent age from a single image without facial landmarks.* Retrieved on May 8, 2021, from https://data.vision.ee.ethz.ch/cvl/publications/papers/articles/eth_biwi_01299.pdf.

[7] Allyn, B. (2020). *Amazon Halts Police Use of Its Facial Recognition Technology.* Retrieved on May 8, 2021, from https://www.npr.org/2020/06/10/874418013/amazon-halts-police-use-of-its-facial-recognition-technology.

[8] Allyn, B. (2020). *IBM Abandons Facial Recognition Products, Condemns Racially Biased Surveillance.* Retrieved on May 8, 2021, from https://www.npr.org/2020/06/09/873298837/ibm-abandons-facial-recognition-products-condemns-racially-biased-surveillance

Though age prediction and facial prediction are different tasks, age is one attribute we use when describing someone to another person. For instance, “She’s a 32-year old girl with long black hair and big brown eyes.” Another example is the identification of a crime suspect using age, e.g. “The suspect is described as a male, approximately 25-35 years old.”

The Technology

The ideas that enable computer image classification and eventually facial recognition have been around for decades. However, recent advancements in open-source software, computing power, scientific research advancement, improved model architecture, and perhaps most importantly, increased number of labeled data, have dramatically increased the performance of image classification models. Nowadays, we expect that a computer can recognize a famous person instantly through Google Lens or a Google Reverse Image Search.

Today, we have access to millions of images organized in public datasets which are labeled with the person’s birth date and the date the image was taken.

---

2. APPROACH

2.1 Data Acquisition and Wrangling

Dataset

The problem of predicting age from a photo is not a novel problem. In fact, there are numerous massive public datasets of faces with the requisite metadata to begin this experiment. Although web-crawling fresh data could be tedious fun, let’s begin with a head start.

In this project we used an extensive public dataset from the DEX research paper related to an image classification competition called LAP Challenge (ChaLearn Looking At People). The dataset “IMDB-WIKI– 500k+ Face Images with Age and Gender labels” was “the largest publicly available dataset of face images with gender and age labels for training” at the time of its publication (2015).[9] The dataset, of over 500,000 face photos, was acquired from IMDb.com and Wikipedia.com.

[9] Rothe, R., Timofte R., Van Gool, L. (2015). *IMDB-WIKI – 500k+ Face Images with Age and Gender Labels.* Retrieved on May 8, 2021, from https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/.

Preprocessing

The half-million image dataset can be downloaded as raw images or as cropped-and-centered images with a standard 40% margin around each face. Face detection was enabled by the work of Mathias, et al., as summarized in their paper, "Face Detection Without Bells and Whistles."[10] We resolved to use the preprocessed cropped images to focus directly on the age classification problem, since face detection is an entirely different and complex domain. All of the cropped color images used in this project are 224 x 224 pixels.

Metadata

The metadata for the images is in a Matlab '.mat' format with the following features: **date of birth** (dob), **year the photo was taken** (photo_taken), **location of file in folders** (full_path), **gender** (0:female, 1:male, NaN:unknown), **name of subject** (name), **location of the face in the image** (face_location from the preprocessing phase), **face detection score** (face_score - rating the confidence a face is detected), **second face detection score** (second_face_score - in the case where there were more than one face in the image). Additionally, the IMDb dataset has 2 more features: celebrity name (celeb_name) and celebrity unique number (celeb_id)—see Table 1.

dob	photo_taken	full_path	gender	name	face_location	face_score	second_face_score	celeb_names	celeb_id
713112	1995	[53/nm0000553_rm2838728192_1952-6-7_1995.jpg]	1.0	[Liam Neeson]	[[905.6965252309108, 94.72990002368314, 1112.1...	3.160553	NaN	NaN	11811
713980	2003	[87/nm0000487_rm1350473984_1954-10-23_2003.jpg]	1.0	[Ang Lee]	[[112.36024808555025, 40.551660030553656, 183....	1.961858	NaN	NaN	1069
717041	2005	[94/nm0005094_rm3393174016_1963-3-11_2005.jpg]	0.0	[Alex Kingston]	[[414.6645228838918, 60.09493184055597, 590.94...	4.264515	1.414976	NaN	461
709153	1969	[88/nm1405388_rm3361982464_1941-8-5_1969.jpg]	1.0	[Don Rich]	[[1, 1, 2012, 3000]]	-inf	NaN	NaN	5268
721982	2004	[81/nm0842081_rm3257383168_1976-9-19_2004.jpg]	0.0	[Alison Sweeney]	[[1, 1, 400, 600]]	-inf	NaN	NaN	650

Table 1. Metadata sample of Dataset

Data Verification

After downloading the dataset with the metadata, we needed to verify that the data is valid. We used Pandas to perform data parsing. We calculated the image age of each subject by subtracting the subject's year of birth from the year the photo was taken. We then plotted a histogram for the age distribution (see Figure 1). In the histogram, we see a class imbalance, a right-tailed distribution with more observations in the 20–60 age

[10] Mathias, et al. (n.d.). *Face Detection without Bells and Whistles* (Tech.). Retrieved May 8, 2021, from https://projet.liris.cnrs.fr/imagine/pub/proceedings/ECCV-2014/papers/8692/86920720.pdf.

range. To avoid training bias, it is important to balance the classes of ages. Ideally, all of the classes of interest would have the exact same number of observations.

Class Balancing

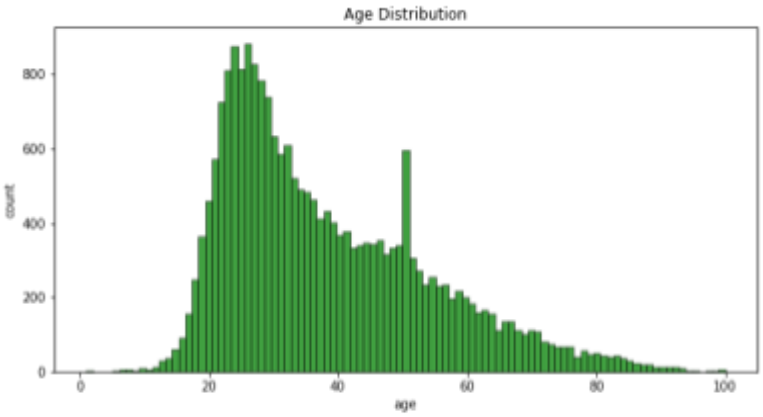


Figure 1. *Age Distribution with class imbalance*

Since the younger (under 8) and the older people (over 80) are not well-represented, we decided to simplify the problem by making predictions within a smaller range of values. Eight (8) iterations were made to create a balanced model with an age range of 0-100, with unacceptable average prediction error (11 or more years from the true age). Hence, we decided to focus on predicting only for the age range 8-80 (73 classes). Working with an age-limited dataset allows us to have a balanced class distribution. To balance the classes precisely we downsampled the overrepresented classes to match the underrepresented classes.

Data Cleaning

The preprocessing facial detection results in a "face score," which we used to filter the dataset, by setting a minimum threshold. In other words, we are only concerned with images where there is a strong face detection score. As mentioned previously, the face detection score ("face score") is a rating that a face is detected, included in the dataset's metadata. Additionally, we opted to eliminate images with a second face detected ("second face score"), to exclusively focus on images where only one face is detected.

Images to Vectors of Pixels

At this point, we have a dataset of thousands of square color images, each composed of 224 x 224 pixels, totaling 50,176 pixels per color channel. Each pixel intensity is represented by a value 0-255, for each channel. There are 3 channels of color— RGB: red, green, blue. Essentially, there is a sandwich of 3 layers of color stacked on top of each other to form a single RGB color image. The product of 3 channels with 50,176 pixels each is 150,528 pixels per image. The classical machine learning approach to representing images with dimensions r x c x k (r rows, c columns, and k channels) calls for creating a 1-dimensional vector of length N = r * c * k. Therefore, we need to extract the pixel values from each image and create a matrix 'X' of observations that has one row per image, and N columns. The extraction of pixels from the images to create the

rows of X is a compute-intensive operation. After the extraction, we saved X as a compressed Numpy array.

TensorFlow & Keras

We used the open-source machine learning library TensorFlow to create our deep learning model. TensorFlow was developed by Google Brain and formally released publicly in 2017. In this project, we used TensorFlow 2.0 released in 2019. Additionally, we used the Keras library to interface with TensorFlow. To use TensorFlow, we reshaped our matrix X into a tensor-like Numpy array. The expected format is 'Rank 4,' ordered as: batch size, width, height, channels (or features). (see Figure 2.)

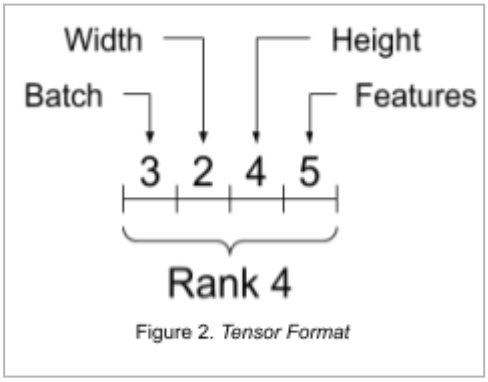


Figure 2. *Tensor Format*

2.2 Data Exploration

Next, we explored the images and the tabular metadata.

Years the Photos Were Taken

We plotted a histogram of the years that the photos were taken. As we might expect, most of the images in the dataset were created in the last 20 years, coinciding with the boom of digital imagery and the explosion of digital data creation. (see Figure 3)

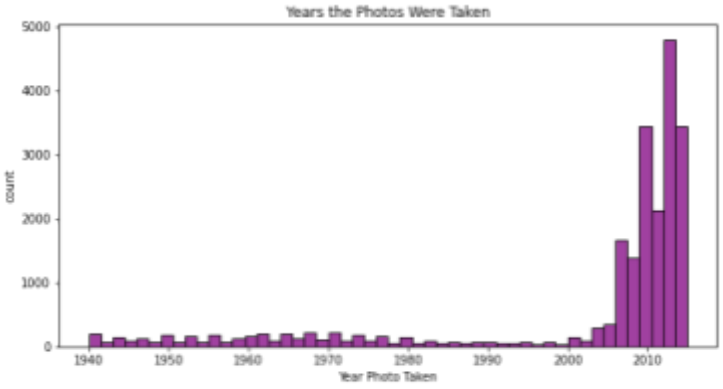


Figure 3. *Years the Photos Were Taken.*

Face Scores Distribution

We plotted a histogram of the face scores. Note that we set the minimum face score at 3.0 to ensure a strong face detection. The distribution is cropped at the low end and has a smooth right tail. (see Figure 4)

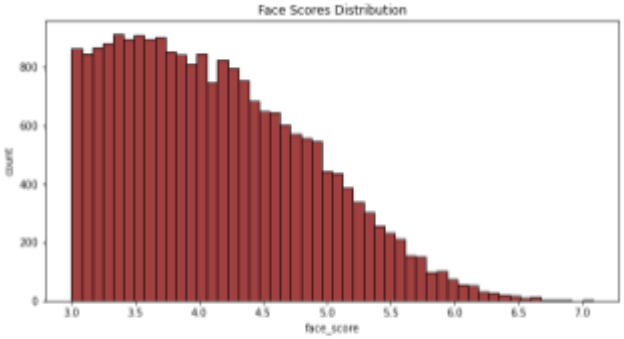


Figure 4. *Face Scores Distribution.*

‘Average Face’ Images

It is interesting to think about the concept of an "average face" from our dataset. What would that look like? Well, since we already have our face image dataset, we can aggregate the mean (average) of the arrays of images and create "average face" images with various subsets. We tried this with 2, 5, 20, 100, and 1,000 face images for fun and to see if any clear patterns emerged. Similarly, National Geographic created a composite face from 190,000 overlayed images to create an 'everyman' concept image.[11]

Image 3. *Average of 1000 Female Images.*

Image 4. *Average of 1000 Male Images.*

Image 5. *Average of Age 60−69 Images.*

In the above composite images (Image 3-5), we can see that there *are* patterns. The image on the left is the average of 1,000 female images from the dataset and it is noticeable that the image shows an appearance of longer hair and the lips appear fuller and more pinkish than the male image. The image in the center is composed of 1,000 images of males and its distinctive features include shorter hair than the composite image of a female. Additionally, we can observe a thicker eyebrow and darker lips than the female image. In the image on the right (Image 5), composed of male and female images aged 60-69, the hair appears lighter with less forehead coverage and the face is rounder.

Next, let’s go deep into the machine learning section of the data science method.

---

[11] Strassman, M. (2011). *Meeting Earth’s most typical person.* Retrieved on May 8, 2021, from https://www.cbsnews.com/news/meeting-earths-most-typical-person/.

2.3 Modeling

Convolutional Neural Networks (CNNs)

CNNs are deep neural networks that are modeled on the biological visual cortex of animals. Deep neural nets use hierarchical layers of artificial neural networks to execute machine learning. CNNs are used for image recognition, image classification, image segmentation, natural language processing, financial time series forecasting, etc. CNNs consistently achieve state-of-the-art results in many of these tasks and in particular, image classification.[12]

Brief History

Modern CNN work leapt forward in the 90s with the research of Yann LeCun, et al., titled "Gradient-Based Learning Applied to Document Recognition," which was used to effectively classify handwritten numbers.[13] Fast-forward to 2012, a CNN architecture called AlexNet (created by AI legends Alex Krizhevsky, Ilya Sutskever, and Geoff Hinton) won the annual ImageNet Large Scale Visual Recognition Challenge by a landslide (>10% points). This was a milestone moment in the development of computer vision and deep learning as the impressive abilities of a CNN combined with GPUs (Graphical Processing Units) were demonstrated on the world stage.

CNN Mechanics

How do CNNs work for image classification? Well, that's a really complex topic and here's a quick overview.

Generally speaking, a CNN consists of multiple layers in three categories: (1) convolutional layers (followed by an activation function); (2) pooling layers; and (3) fully connected layers. Why convolution? There is a small square filter (3x3 pixels or 5x5 pixels, for example) that convolves - moving systematically from left-to-right along multiple rows of pixels, then left-to-right on the next set of rows of pixels performing element-wise multiplication against certain filter values and summing the values into a 'convolved feature.'

At the start of the network, the filters are learning to detect simple features such as a diagonal line. Later network filters are able to discern more complex (higher-level) features such as an eyeball shape. Next, there is a 'max-pooling' layer that

[12] *Wikipedia: Convolutional Neural Network*. (n.d.). Wikipedia. Retrieved May 8, 2021, from https://en.wikipedia.org/wiki/Convolutional_neural_network.
[13] LeCun, Y., et al. (1998). *Gradient-Based Learning Applied to Document Recognition.* Retrieved on May 8, 2021, from http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf.

down-samples the convolved features, reducing the overall amount of data. This sequence of convolution and pooling can be repeated any number of times depending on the specific use case. In fact, some 'deep' CNNs are composed of over 100 layers such as Microsoft's 2015 ImageNet challenge winner.[14]

Model Architecture

For our case, we started with a 22-layer model architecture called 'VGG-Face' from the Visual Geometry Group, at the University of Oxford.[15] Our implementation was inspired by "Apparent Age and Gender Prediction in Keras" by Sefik Ilkin Serengil.[16] In Figure 6 below, we can see the VGG-Face model interplay between convolution layers and max-pooling layers. Eventually, there are fully connected layers ending with a softmax layer which outputs marginal probabilities of each class using a sigmoid function.

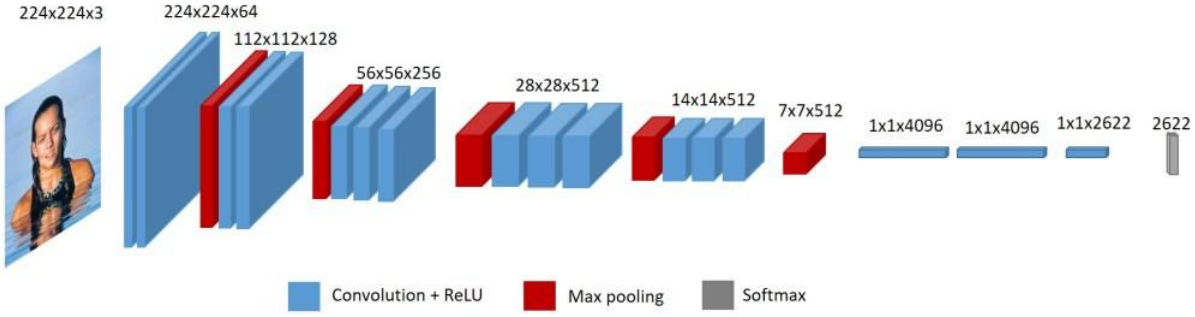


Figure 6. VGG-Face Model Overview[17]

2.4 Model Application

Performance Metrics

We are modeling the age prediction task as a classification task, followed by a regression task. The classification task produces a set of probabilities that estimate the likelihood that the input belongs to a specific age class. The regression task takes the maximum probability and produces a number that is associated with the corresponding class. The *de facto* performance metric for age prediction in the related literature and competitions is Mean Absolute Error (MAE).[18] MAE is an average (arithmetic mean) of the absolute difference between each predicted regression value from the ground-truth labels (actual

[14] He, Kaiming; Zhang, Xiangyu; Ren, Shaoqing; Sun, Jian (2016). *Deep Residual Learning for Image Recognition*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR): 770–778.
[15] Parkhi, O., Vedaldi, A., Zisserman, A. (n.d.). *VGG Face Descriptor*. Retrieved on May 10, 2021, from https://www.robots.ox.ac.uk/~vgg/software/vgg_face/.
[16] Serengil, S. (2019). *Apparent Age and Gender Prediction in Keras*. Retrieved on May 10, 2021, from https://sefiks.com/2019/02/13/apparent-age-and-gender-prediction-in-keras/.
[17] Serengil, S. (2019). *Deep Face Recognition with Keras*. Retrieved on May 10, 2021, form https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/
[18] Rothe, R., Timofte, R., Van Gool, L. (2016). *Deep expectation of real and apparent age from a single image without facial landmarks*. Retrieved on May 10, 2021, from https://data.vision.ee.ethz.ch/cvl/publications/papers/articles/eth_biwi_01299.pdf.

age). In addition to the MAE, we will also track mean absolute percentage error (MAPE)[19] , test set classification accuracy, and test set categorical cross-entropy classification loss. Categorical cross-entropy computes the classification error between predictions and labels. Ideally, we want to minimize this loss as we iterate through our batches and models. In this project, CNN hyperparameters are optimized with respect to classification metrics.

Constants

We fixed some variables with constants to compare different models. We set the number of epochs at 200, batch size at 128. One epoch is completed when the entire dataset has been passed forward and backward through the neural network. A batch is a defined subset of the entire dataset. A batch size is the total number of training examples in a single batch.[20] Additionally, we consistently used a 70% training set and 30% test set split, stratified by target classes with 3,525 training set examples and 1,512 test set examples.

We started by creating a baseline model to assess performance benchmarks. First, let's look at model results and at the end, we will evaluate their relative performance.

2.4.1 Model Details: CNN 1

Baseline Model (CNN1) Details

Our first CNN used VGG-Face "out of the box," without tweaking the model. The only thing that must be changed is the number of output neurons. The model was built to predict 2,622 classes. We reduced the number of output neurons to match our 73 target classes.

Optimizer

The Baseline model uses a Stochastic Gradient Descent (SGD) Optimizer with the following settings: learning_rate=0.1, decay=1e-6, momentum=0.9, nesterov=True.

Test Set Model Performance Metrics

Table 2. CNN 1 Performance Metrics

Model Performance Metrics				
	MAE	**MAPE**	**Test Accuracy**	**Test Loss**
CNN1	**11.22**	**44.43%**	**0.04696**	**3.92303**

[19] MAPE = Mean Absolute Percent Error = Mean of [(Actual - Predicted) / Actual]

[20] Sharma, Sagar (2017). Epoch vs. Batch Size vs Iterations. Towards Data Science. Retrieved on May 10, 2021 from https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9

Performance Visualization: CNN 1

1) Training Loss / Validation Loss

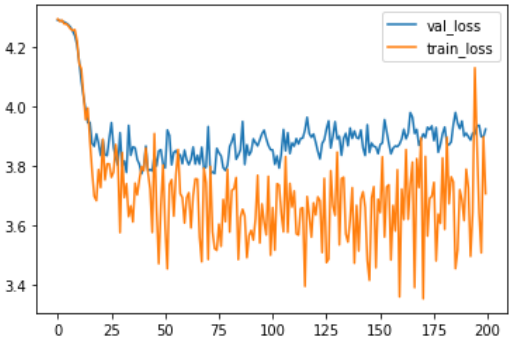


Figure 7. *CNN 1 Loss*

2) Training Accuracy / Validation Accuracy

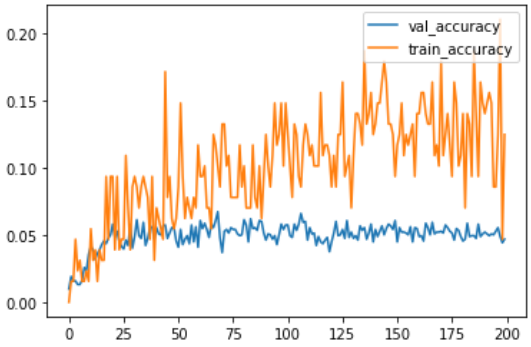


Figure 8. *CNN 1 Accuracy*

3) Jointplot: Actual Age / Predicted Age

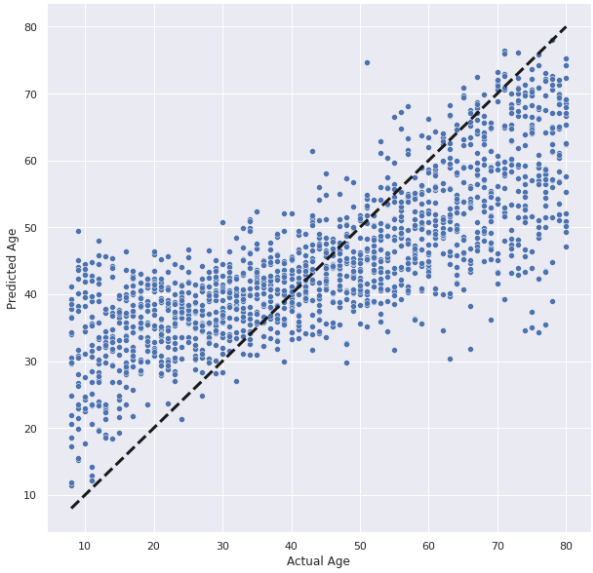


Figure 9. *CNN 1 Jointplot*

4) Confusion Matrix: Actual Age / Predicted Age

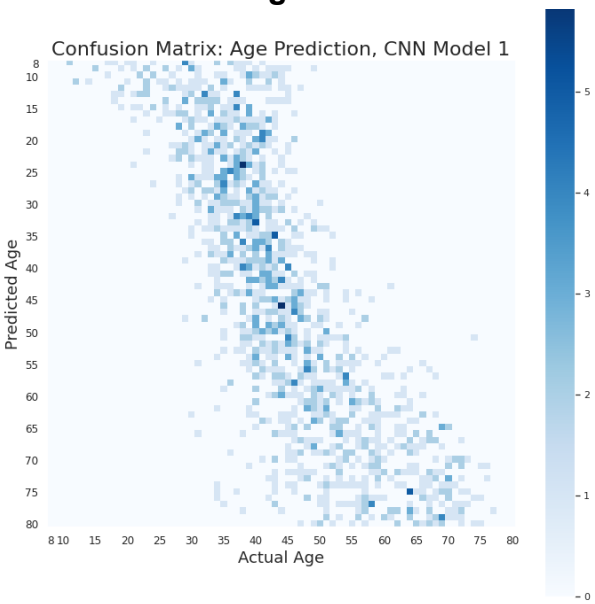


Figure 10. *CNN 1 Confusion Matrix*

Observations

The Baseline model has an MAE of 11.22, MAPE of 44.43%, Test Set Accuracy of 4.69%, and Test Set Loss of 3.92303. These numbers serve as our benchmark from which to improve the model. In figure 7 and figure 8, we see a the training set diverge from the validation set on accuracy and loss, possibly indicating overfitting.

2.4.2 Model Details: CNN 2

Our second CNN model will use the VGG-Face with an Adam (Adaptive Moment Estimation) optimizer.

Optimizer

CNN 2 model uses an Adam Optimizer with the following settings: learning_rate=0.0007, beta_1=0.9, beta_2=0.999, epsilon=1e-07.

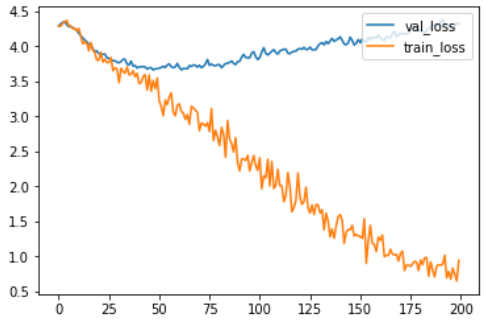
Test Set Model Performance Metrics

Model Performance Metrics				
	MAE	**MAPE**	**Test Accuracy**	**Test Loss**
CNN 1	11.22	44.43%	0.04696	3.92303
CNN 2	**7.94**	**25.50%**	**0.08466**	**4.31565**

Table 3. CNN 2 Performance Metrics.

Performance Visualization: CNN 2

5) **200 Epochs: Training Loss / Validation Loss**



6) **200 Epochs: Training Accuracy / Validation Accuracy**

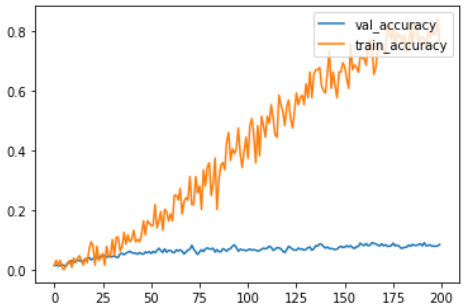


Figure 11. *CNN 2 Loss*

Figure 12. *CNN 2 Accuracy*

7) Jointplot: Actual Age / Predicted Age

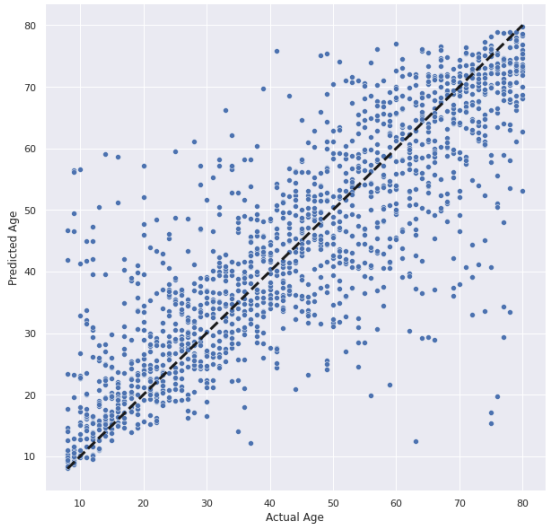


Figure 13. *CNN 2 Jointplot*

8) Confusion Matrix: Actual Age / Predicted Age

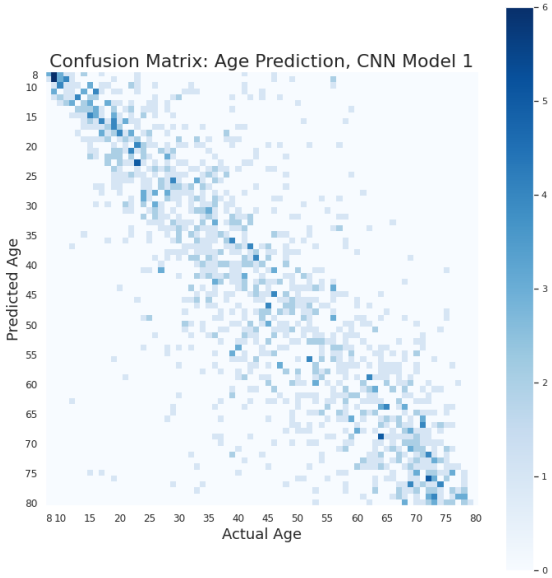


Figure 14.*CNN 2 Confusion Matrix*

Observations

The CNN 2 model has an MAE of 7.94 and MAPE of 25.50% a vast improvement from the baseline model on both metrics. The Test Set Accuracy is up to 8.466%, however the Test Set loss increased which is not a good sign. Observing figure 11 and figure 12 we see an early divergence of the training set from the test set which may indicate more extreme overfitting than the baseline model.

2.4.3 Model Details: CNN 3

CNN 3 model will use the VGG-Face architecture with an Adam (Adaptive Moment Estimation) optimizer and pretrained weights to leverage transfer learning. Transfer Learning utilizes knowledge gained from another problem and applies it to another related problem.[21]

Optimizer

CNN 3 model also uses an Adam Optimizer with the following settings:
learning_rate=0.0007, beta_1=0.9, beta_2=0.999, epsilon=1e-07.

[21] West, Jeremy; Ventura, Dan; Warnick, Sean (2007). *A Theoretical Foundation for Inductive Transfer.* Spring Research Presentation. Brigham Young University, College of Physical and Mathematical Sciences. Archived from the original on 2007-08-01.

Test Set Model Performance Metrics

Model Performance Metrics				
	MAE	**MAPE**	**Test Accuracy**	**Test Loss**
CNN 1	11.22	44.43%	0.04696	3.92303
CNN 2	7.94	25.50%	0.08466	4.31565
CNN 3	**7.78**	**25.23%**	**0.06019**	**3.65387**

Table 4. CNN 3 Performance Metrics.

Performance Visualization: CNN 3

9) 200 Epochs: Training Loss / Validation Loss

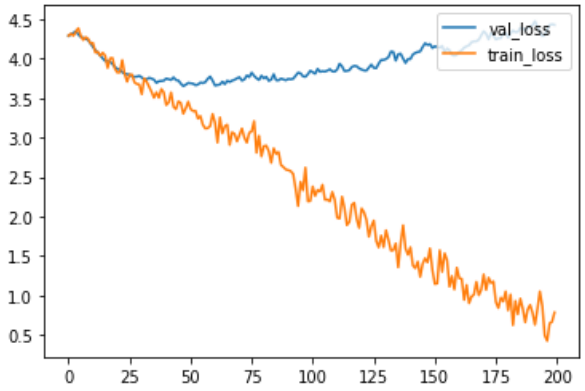


Figure 15. *CNN 3 Loss*

10) 200 Epochs: Training Accuracy / Validation Accuracy

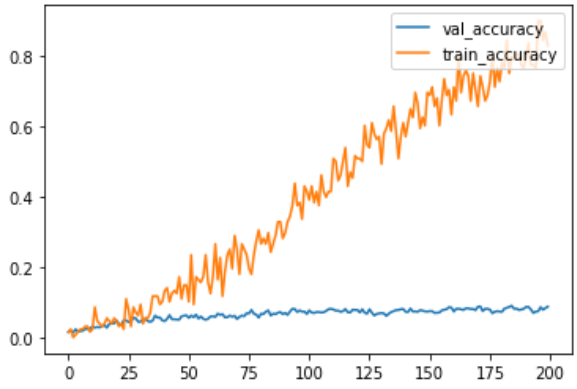


Figure 16. *CNN 3 Accuracy*

11) Jointplot: Actual Age / Predicted Age

12) Confusion Matrix: Actual Age / Predicted Age

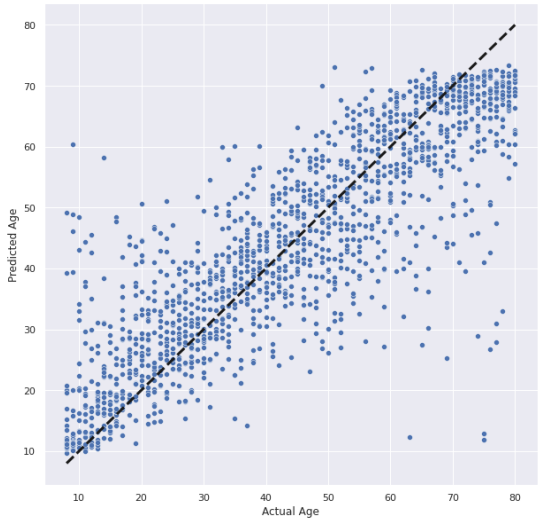


Figure 17. *CNN 3 Jointplot*

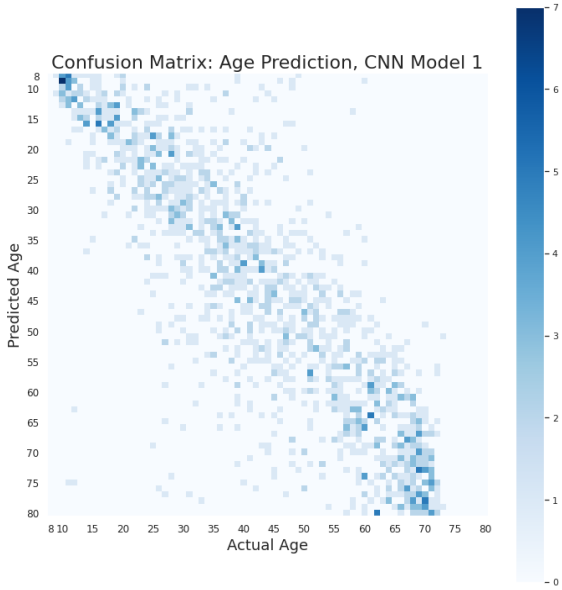


Figure 18. *CNN 3 Confusion Matrix*

Observations

The CNN 3 model has a slightly improved MAE of 7.78, MAPE of 25.23% and Test Set Loss of 3.65387. However, the Test Set Accuracy has decreased to 6.019%. Again, figures 15 and 16 indicate overfitting.

2.4.4 Model Details: CNN 4

CNN 4 model will use the VGG-Face architecture with an Adam (Adaptive Moment Estimation) optimizer and pretrained weights and data augmentation. Data augmentation randomly transforms the input data to add diversity to the training data set to hopefully increase model generalizability.

Optimizer

CNN 4 model also uses an Adam Optimizer with the following settings:
learning_rate=0.0007, beta_1=0.9, beta_2=0.999, epsilon=1e-07.

Test Set Model Performance Metrics

Model Performance Metrics				
	MAE	**MAPE**	**Test Accuracy**	**Test Loss**
CNN 1	11.22	44.43%	0.04696	3.92303
CNN 2	7.94	25.50%	0.08466	4.31565

CNN 3	7.78	25.23 %	0.06019	3.65387
CNN 4	**7.54**	**24.74 %**	**0.06878**	**3.59017**

Table 5. CNN 4 Performance Metrics

Performance Visualization: CNN 4

13) 200 Epochs: Training Loss / Validation Loss

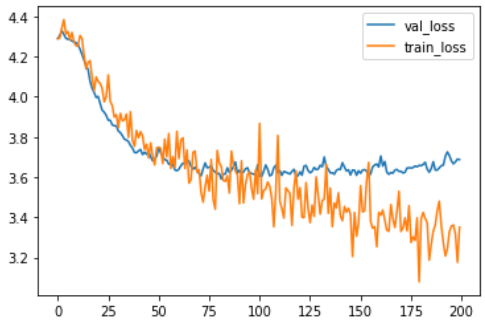


Figure 19. *CNN 4 Loss*

14) 200 Epochs: Training Accuracy / Validation Accuracy

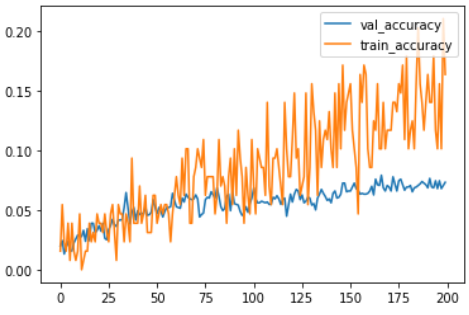


Figure 20. *CNN 4 Accuracy*

15) Jointplot: Actual Age / Predicted Age

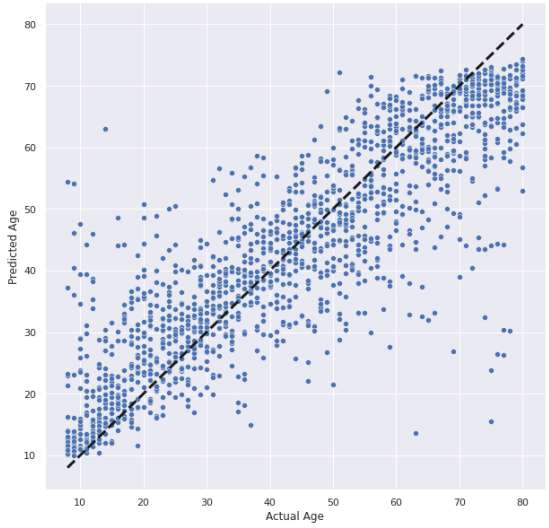


Figure 21. *CNN 4 Jointplot*

16) Confusion Matrix: Actual Age / Predicted Age

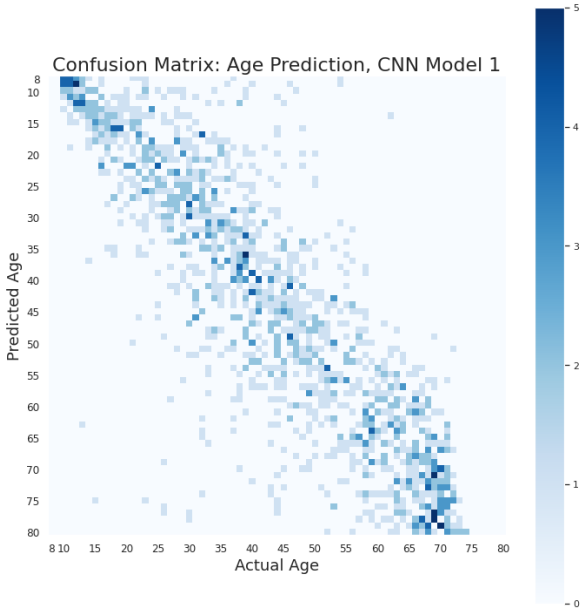


Figure 22. *CNN 4 Confusion Matrix*

Observations

The CNN 4 model has an improved MAE of 7.54, MAPE of 24.74% and Test Set Loss of 3.59017. The Test Set Accuracy has improved to 6.878%. In figure 19 we can see the training set accuracy and the test set accuracy moving together until around 100 epochs. Similarly, in figure 20 we can see the training set loss and the test set loss moving together until around 100 epochs. These are good signals that the model is not overfitting. We selected this as our "best" model to solve our age prediction problem.

---

3. Findings

Model Summary

After a progression of experiments, the models have become incrementally more complex. Next is a quick summary of the differences between the models and their performance metrics.

CNN 1 = VGG-Face 'out of the box' with SGD optimizer
CNN 2 = VGG-Face with Adam optimizer
CNN 3 = VGG-Face with Adam optimizer with pretrained weights
CNN 4 = VGG-Face with Adam optimizer with pretrained weights & data augmentation

Model Performance Metrics				
	MAE	**MAPE**	**Test Accuracy**	**Test Loss**
CNN 1	11.22	44.43%	0.04696	3.92303
CNN 2	7.94	25.50%	**0.08466**	4.31565
CNN 3	7.78	25.23 %	0.06019	3.65387
CNN 4	**7.54**	**24.74 %**	0.06878	**3.59017**

Table 6. Performance Metrics Comparison

Model Merits Comparison

In the Model Performance Metrics table above (Table 6), we can see that CNN 4 had the best MAE, MAPE, and test set loss. The model is based on a VGG-Face architecture using an Adam optimizer with pretrained weights & data augmentation. Adam

optimization is used on all the models tested except the baseline model (CNN 1), which uses Stochastic Gradient Descent (SGD). The baseline model has the advantage of being easy to implement. However, in CNN2 as we change the optimizer to Adam, we observed a lower MAE. By loading pretrained weights in CNN 3 and 4, we took advantage of a highly pretrained model that we can build upon. Finally in CNN 4, we observed a slight performance boost using data augmentation to transform the images by random flipping, rotating, and zooming. The increase in training data diversity can improve model generalizability.

Best Model

The best model we have trained is CNN 4. It takes advantage of a pretrained network that uses an Adam optimizer and data augmentation.

Business Results

CNN 4 achieved an MAE of 7.54. In terms of creating an age predictor, this means that the average age 'guess' is 7.54 years away from the actual value. While it could be useful to get a ballpark estimate of an age, our client might desire a smaller prediction error before creating a product. The estimator can be improved by training with more labeled data and model fine-tuning.

APP Demo

We created an "App" demo in our Google Colab notebook which can be fed an image web address (URL) and uses CNN 4 to output an age prediction. We hard-coded the "Actual Age" based on research about the photo date and the celebrities' birthdays. The following images are the results which exhibit a wide range of errors demonstrating that the model shows promise, but needs additional optimization.

Image 6. *Image of Dwayne Johnson.*

Predicted Age: 51

Actual Age: 48

Image 7. *Image of Celine Dion.*

Predicted Age: 61

Actual Age: 52

Image 8. *Image of Britney Spears.*

Predicted Age: 41

Actual Age: 34

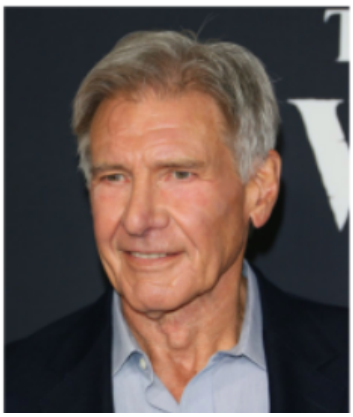


Image 9. *Image of Harrison Ford.*

Predicted Age: 66

Actual Age: 70

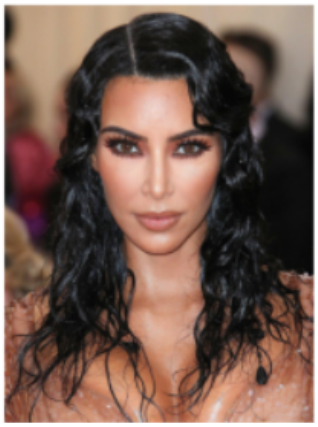


Image 10. *Image of Kim Kardashian.*

Predicted Age: 34

Actual Age: 37

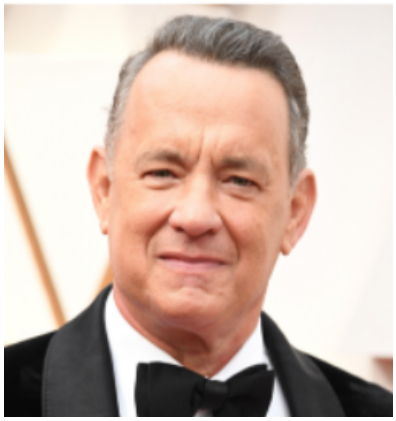


Image11. *Image of Tom Hanks.*

Predicted Age: 64

Actual Age: 63

Getty Images

---

4. Conclusions & Future Work

Conclusions

Deep Learning Convolutional Neural Networks are suitable for age prediction. The task of 'learning' the features associated with certain ages and then predicting ages on new images, is quite complex.

Remarkably, we were able to create a functional image-based age predictor model using mostly open-source software, free tools and a free public dataset. Python, TensorFlow, Keras, Google Colab, etc., enable a very exciting world of AI innovation. We are currently at the relative infancy of the AI revolution and it's a very exciting time to become exposed to and learn data science.

Open Questions & Answers

1. **Can we use public datasets and open-source software to create an age predictor with reasonable accuracy?**

We were able to accomplish reasonable accuracy predicting ages using only public datasets and open-source software. The Mean Absolute Error (MAE) for our model is 7.54 years and the Mean Absolute Percentage Error is 24.74%.

2. **What are the major limitations and challenges when attempting to estimate age?**

The major challenges and limitations when estimating age is the possibility that people can look older or younger than their age in photos due to myriad reasons such as makeup, filters, image augmentations, etc. The practical way to address these challenges is by simply feeding more labeled training examples into the model.

3. **Are there any systemic biases in public datasets? How to overcome any biases?**

Yes, there are systemic biases in many public datasets. Public datasets which acquire images of famous people will have inherent biases. The main bias in our dataset was the highly imbalanced age classes. There were simply not enough images of people under the age of 8 and over the age of 80. To minimize bias, we opted to reduce the number of classes to predict from 101 age classes to 73 age classes.

Future Work

To further develop this work, we'd like to start from the data acquisition phase and crawl the internet for more faces of the underrepresented age classes of the younger and older people, resulting in a less age biased dataset. Alternatively, we could find a dataset or combine datasets to create a more balanced distribution of age classes. We'd like to build a predictor with an age range of 0-100 as we initially set out to do. We'd also like to experiment with other architectures with pretrained weights to improve performance.

---

5. Recommendations

- We recommend to our client that we create the Beta version of the app, since our age predictor using CNN4 offers promising performance metrics. We recommend releasing the Beta with the disclaimer that this is an early iteration that will become optimized over time.
- We recommend that we focus first on high-quality data acquisition to fuel the deep learning machine. More, balanced, and less biased data may be the number one factor in creating a more robust age predictor.
- We would like to do a survey of other pretrained models and test how well they perform with our age prediction problem.

---

6. Lessons Learned

The following is a list of *personal* lessons I learned as I worked on this project:

- *"Your session crashed after using all available RAM."* in Google Colab is usually a bad sign.
- We are in a very exciting time in history where people can build impressive tools using open-source software.
- Python, TensorFlow, Keras, Google Colab and public datasets can accomplish the seemingly impossible. The rapid development and democratization of AI is incredible and exciting.
- Reading the official documentation is often the fastest route to an answer rather than reading blogs or other tutorials.
- Class imbalance has been one of the most difficult challenges to overcome on this project. Downsampling and data augmentation were both used to address class imbalance.
- Google Colab Pro was a worthwhile investment for me as it allowed me to train with GPUs and high-RAM without running out of RAM (as frequently).
- I think that it is a good practice to record **ALL** experiments even (especially) when they don't work.
- Keep it simple. Beware of feature creep. *"Premature Optimization is the root of all evil."*
- Be kind to yourself and patient with yourself while learning.

---

7. References

1. Rothe, R., Timofte R., Van Gool, L. (2015). *IMDB-WIKI – 500k+ Face Images with Age and Gender Labels.* Retrieved on May 10, 2021, from https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/.
2. Rothe, R., Timofte, R., Van Gool, L. (2016). *Deep expectation of real and apparent age from a single image without facial landmarks.* Retrieved on May 10, 2021, from https://data.vision.ee.ethz.ch/cvl/publications/papers/articles/eth_biwi_01299.pdf.
3. Parkhi, O., Vedaldi, A., Zisserman, A. (n.d.). *VGG Face Descriptor*. Retrieved on May 10, 2021, from https://www.robots.ox.ac.uk/~vgg/software/vgg_face/.
4. Krizhevsky, A., Sutskever, I., and Hinton, G.E. (2012) *ImageNet Classification with Deep Convolutional Neural Networks*. In NIPS, pages 1106–1114, 2012.
5. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., and Jackel, L.D. (1989). *Backpropagation Applied to Handwritten Zip Code Recognition*. In *Neural Computation*, vol. 1, no. 4, pp. 541–551.
6. Mathias, M., Benenson, R., Pedersoli, M., and Van Gool, L. (2014). *Face Detection without Bells and Whistles*. In Proc. ECCV. Retrieved on May 10, 2021, from https://link.springer.com/content/pdf/10.1007%2F978-3-319-10593-2_47.pdf.
7. Serengil, S. (2019). *Apparent Age and Gender Prediction in Keras*. Retrieved on May 10, 2021, from https://sefiks.com/2019/02/13/apparent-age-and-gender-prediction-in-keras/.
8. Abadi, M., et al. 2016. *TensorFlow: A System for Large-Scale Machine Learning.* In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI'16), pp. 265–283. Retrieved on May 10, 2021, from https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf
9. Chollet, F., & others. (2015). *Keras*. GitHub. Retrieved on May 10, 2021, from https://github.com/fchollet/keras.
10. Van Rossum, G. and Drake Jr, F. L. (1995). *Python Reference Manual*. Centrum voor Wiskunde en Informatica Amsterdam.
11. McKinney, W.. (2010). *Data Structures for Statistical Computing in Python*. In Proceedings of the 9th Python in Science Conference (Vol. 445, pp. 51–56).

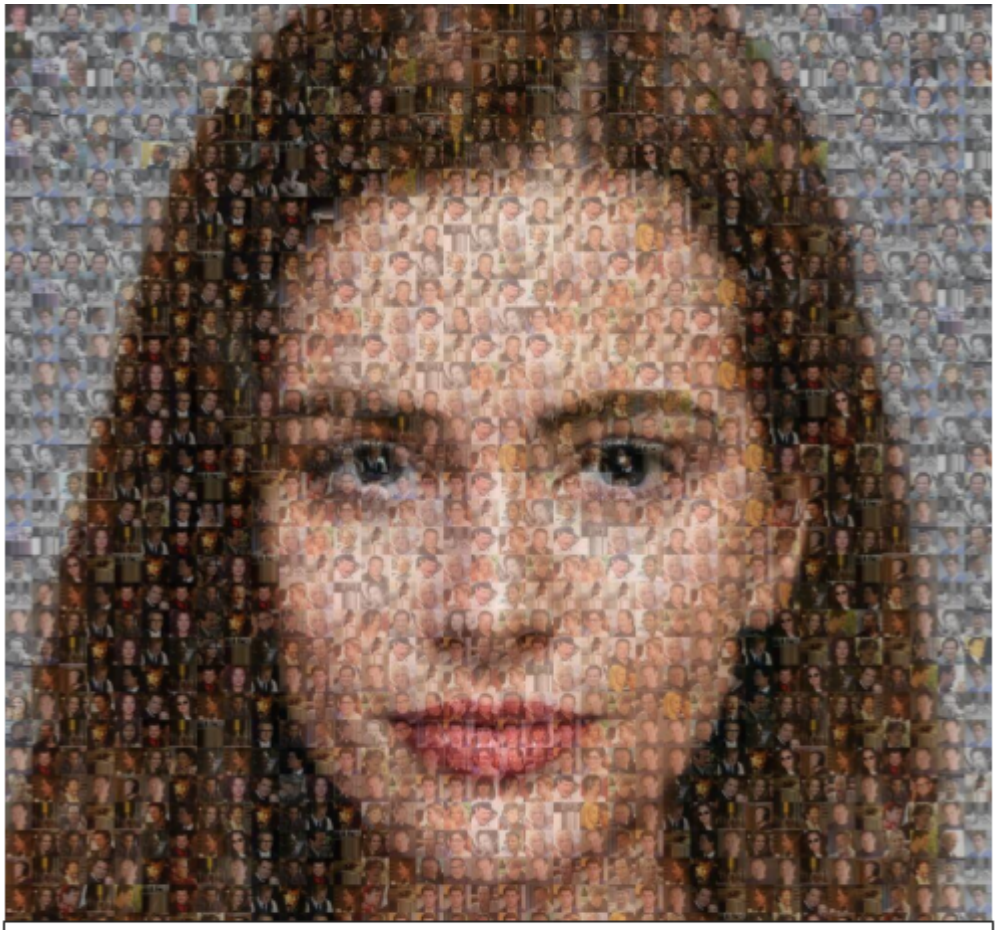
12. Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. (2009). *Imagenet: A Large-Scale Hierarchical Image Database*. In 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255.
13. Kluyver, T., et al. (2016). *Jupyter Notebooks – a publishing format for reproducible computational workflows*. In F. Loizides & B. Schmidt (Eds.), Positioning and Power in Academic Publishing: Players, Agents and Agendas, pp. 87–90.
14. Agarwal, P. (2020). *Age Detection Using Facial Images: traditional Machine Learning vs. Deep Learning*. Towards Data Science. Retrieved on May 10, 2021, from https://towardsdatascience.com/age-detection-using-facial-images-traditional-machine-learning-vs-deep-learning-2437b2feeab2.
15. Parkhi, O., Vedaldi, A., and Zisserman, A. (2015). *Deep Face Recognition.* British Machine Vision Conference. In Xianghua Xie, Mark W. Jones, and Gary K. L. Tam, editors, Proceedings of the British Machine Vision Conference (BMVC), pages 41.1-41.12.
16. Chauhan, Nagesh, (2019). *Predict Age and Gender Using Convolutional Neural Network and OpenCV*. KDnuggets. Retrieved on May 10, 2021 from https://www.kdnuggets.com/2019/04/predict-age-gender-using-convolutional-neural-network-opencv.html
17. Sharma, Sagar (2017). *Epoch vs. Batch Size vs Iterations*. Towards Data Science. Retrieved on May 10, 2021 from https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9
18. West, Jeremy; Ventura, Dan; Warnick, Sean (2007). *Spring Research Presentation: A Theoretical Foundation for Inductive Transfer.* Brigham Young University, College of Physical and Mathematical Sciences. Archived from the original on 2007-08-01. Retrieved May 10, 2021
19. Serengil, S. (2019). *Deep Face Recognition with Keras*. Retrieved on May 10, 2021, form https://sefiks.com/2018/08/06/deep-face-recognition-with-keras/

---

8. Acknowledgements

I'd like to thank my incredible Springboard Data Science Mentor, **AJ Sanchez, Ph.D.** Chief Data Scientist and Principal Software Engineer at Exodus Software Services, Inc., for patiently guiding me along in this project.

Also, my wife is pretty cool. Thank you for your inspiration, Pinky!



Mosaic Created from Dataset Images

Younger: Github