UFCFSN-15-3 Artificial Intelligence for Creative Technology Project Documentation	
Student Name:	Tom Smith
Student Number:	19010377
Project Title:	Classification Network for Determining Age From Faces

Abstract
In this report I will discuss the development of the proposed AI image classifier, and the research that has been undertaken alongside it. The desired outcome is the creation of a neural network that can classify images of people into their respective age groups, ideally to a high degree of accuracy.

Introduction
Artificial intelligence is a topic of controversy and intrigue amongst the media, and with deep learning AI is becoming more and more advanced. Using classifier networks such as MLPs and CNNs, AI models can be trained to perceive information and patterns in images with impeccable accuracy, while the use of GANs allow AI to generate entirely new images given some existing examples to work from, even completely new photorealistic human faces. And that is the direction I am taking this project, faces. Our faces are closely bound to our identity, particularly when interaction between people is concerned, our faces carry the discrete quirks of our lives, our character, and our faces can also betray the sentiment of our words or show the extent of our experiences. For an AI to unravel the complexities of the human face is certainly challenging.

Figure 1 *An image of a photo-realistic human woman, who does not exist. Generated by an AI on thispersondoesnotexist.com using StyleGAN2.*

This project will focus on the development of a classifier to ascertain the age groups of people from an image of their face. I will look into other similar work and the methods used within them to produce an AI network that can achieve this with acceptable accuracy. I chose this as the aim due to the potential for use such a network has in a number of areas. A highly accurate network could be used in security and surveillance to help find and monitor people in crowds, and there have been many implementations of such a network in the mobile phone app industry in the form of filters.

Related Works

There are multiple examples of similar work documented online, by researching into these and the methods used in development I can gain valuable insight and potentially improve the quality of the network produced in this project.

Deep Convolutional Neural Network for Age Estimation based on VGG-Face Model
Z. Qawaqneh et al.

A very similar network was developed by Computer Scientists at the University of Bridgeport. In their documentation they discuss the issues that accompany dealing with facial aging in the context of AI classification, noting the non-uniform nature of human faces in general, and the increased variation when aging is also included as a vital aspect, people do not all age at a proportional rate and thus aiming for 100% accuracy when estimating age from a person's face is entirely unrealistic. The results and findings in their testing of the network reinforced what I had considered when choosing this task, achieving an overall accuracy of 59.9%, this however was shown as a considerable improvement when the model was directly compared with GoogLeNet CNN, which had an accuracy in their database of 45.07%. Knowing this, I feel a minimum accuracy to strive for is at least 50%.

Another potential issue discussed is the threat of overfitting when training the network on a training dataset. This issue was combatted in this case by initially training the network for face recognition on a separate larger database, as there supposedly was no relatively large databases for age estimation. This is something that might prove problematic as the dataset I would likely be working with would be incredibly cumbersome to work with if it were to match the larger sets such as ImageNet, or the Adiance dataset as used in this case.

This network was created using a CNN architecture (as CNNs were said to show significant success in face recognition) and using transfer learning from a pre-trained model as alluded to above, this showed improved results from the network and is likely the method I will pursue in this project.

Age Group Classification using Convolutional Neural Network
M F. Mustapha et al.

Another example of work incredibly similar to the previous example, also utilising a CNN. The justification for the method used was much the same as other examples online, CNNs have gained a lot of attention as they can learn 'compact and discriminative feature representation from large-scale data'. However the results gained using this network were less helpful due to what I consider an error in the classification of the images. The images were split into classes labelled 'early adulthood' or 'early adolescence' etc. These are poor examples of classes as the names are inherently subjective, and the bias of the person responsible for sorting the dataset becomes a decisive factor in the performance of the network itself, for it is impossible to definitively state a universal standard for when 'early adolescence' ends and 'late adolescence' begins.

Dataset Collation

After identifying this error I realised I would need classes that were objectively undeniable and rigid in their borders, as such I settled on separating each class by decade. The new classes were: <20, 20s, 30s, 40s, 50s, and 60+. Now to gather data for this new structure I had to know the ages of the people in each picture so that I can classify them for training the network, to ensure this was the case I decided to find pictures of famous celebrities as they had publicly available birth dates. I then, in the method outlined for the first dataset, collected 100 images of celebrities from each age range (this is why the first class is <20, the number of usable images of celebrities under the age of 10 were insufficient for the dataset). The dataset was then split into 3 sections, training, testing & validation (split 70 : 15 : 15 respectively).

There were publicly available datasets I could have used, however they often were far too large to me used on my machine, with disk speed being a major obstacle when frequently manipulating and moving folders with 100000+ images. Thus, I chose to use the smaller manually curated dataset I described above.

Each image used in the dataset was hand selected by myself so that I can omit any unsuitable images – those with poor quality or that obscure the face of the subject – and ensure the are no glaring colour / lighting discrepancies.

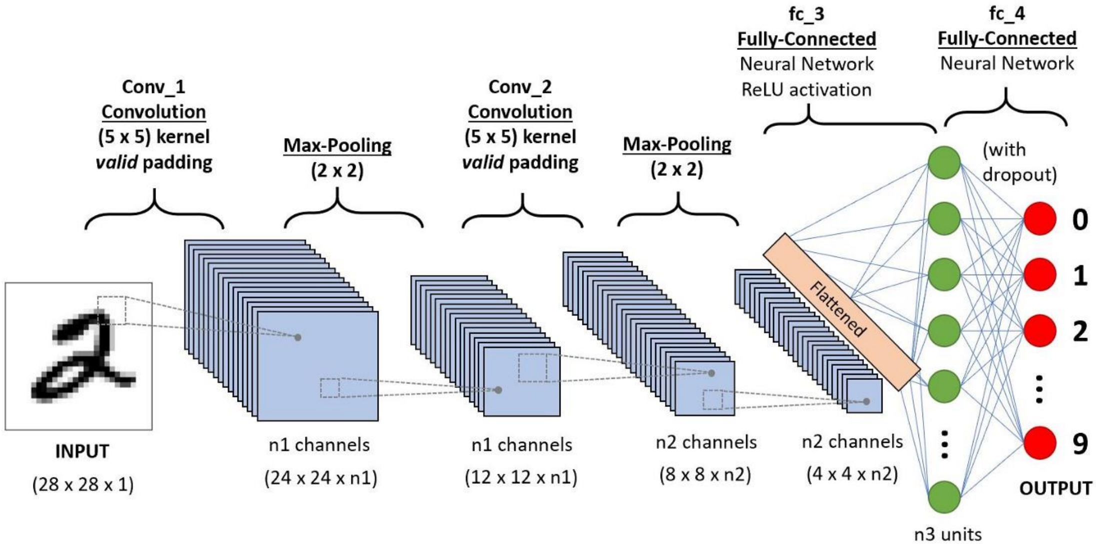
Methods used

The project and codebase is hosted and run on Google Collaborate, which is why the larger datasets were also an obstacle, as I had limited runtime and GPU usage available on the free version of Collaborate, meaning the hours required to train on a large dataset far exceeded what was feasible in my development environment. The codebase uses the Numpy library for mathematical operations, Pytorch as the ML framework, and matlibplot to visualise results. Collaborate was chosen because of its easy link to google drive, as well as the #imports managing dependencies.

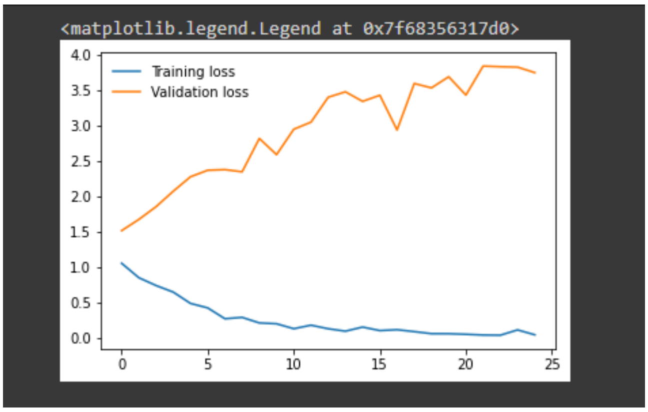
Techniques used

- Multilayer Perceptron
- Convolutional Neural Network
- Transfer Learning
- Data Augmentation

Convolutional Neural Network (CNN)



The architecture used is the CNN architecture as was the case in the other works I highlighted, I had also tested a Multilayer Perceptron (MLP) however I saw far worse results than the CNN. CNNs are a type of artificial neural network commonly applied to images and are named as such due to the convolution filters that take input images and generate feature maps. They function by taking in an image and then assign weights and biases to certain features it identifies in the image, allowing it to differentiate and classify images of the same or similar subject. My initial implementation of this architecture did bring minor improvements but still resulted in low accuracy values, as shown below:




```
Test Loss: 1.920107

Test Accuracy of     0: 50% (10/20)
Test Accuracy of     1: 30% ( 6/20)
Test Accuracy of     2: 20% ( 4/20)
Test Accuracy of     3: 50% (10/20)
Test Accuracy of     4: 25% ( 5/20)
Test Accuracy of     5: 40% ( 8/20)

Test Accuracy (Overall): 35% (43/120)
```

Transfer Learning

Transfer Learning involves taking another pre-trained network and adjusting the weights to fine tune the model to a specific task. This is beneficial as it takes an incredibly large dataset to train a network from the ground up, necessary to combat over-fitting.

PyTorch had a variety of models listed, however I chose to use ResNext, a 50 layer model

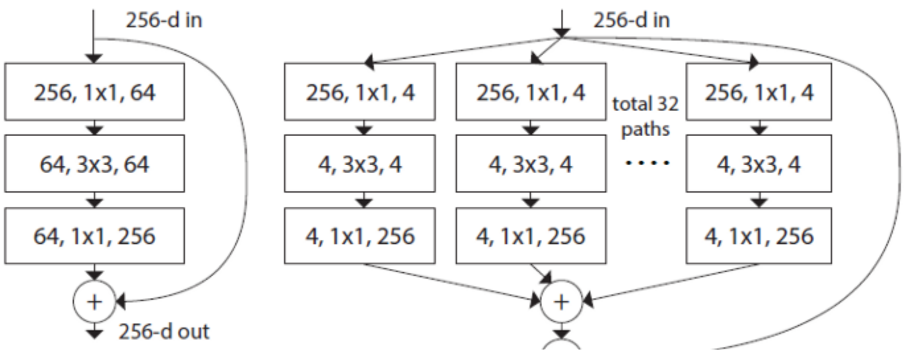


Figure 2: Residual Block in ResNet (Left), A Block of ResNeXt (Right)

freely available online. After finding relative success using ResNet (a different, 18 layer model), I decided to look into ResNext as it was supposed to build upon and improve its predecessor. Unlike ResNet, ResNext inherits from other models such as VGG and Inception to improve accuracy. The inherited features allow it to leverage repeating layers to build a deep architecture model, and it shares hyper-parameters (width and filter sizes) within the residual block.

Using Transfer Learning means that instead of forcing myself into using an enormous dataset and attempting to train the model for tens to hundreds of hours, I can use the pre-trained ResNeXt model and then train it once with my collated dataset. I was still concerned that my dataset might be insufficient size, and so I implemented another technique to combat the risk of overfitting.

Data Augmentation

Data Augmentation is a method of diversifying and augmenting your dataset by applying random transforms to images and then reincorporating them into the dataset, increasing the dataset's size.

```
# TRAINING (with DATA AUGMENTATION)
train_transform = transforms.Compose([transforms.Resize(236),
                              transforms.RandomRotation(5), # RANDOM ROTATION
                              transforms.CenterCrop(224), # CENTER CROP
                              transforms.RandomHorizontalFlip(), # RANDOM FLIP
                              transforms.ToTensor(),
                              transforms.Normalize([0.485, 0.456, 0.406],
                                                   [0.229, 0.224, 0.225])]) # NORMALISE FOR ALL COLOUR CHANNELS (REQUIRED FOR PRETRAINED MODEL)
```

Figure 3: The code in collaborate used when initialising training, random rotations are applied followed by a centre crop to remove any black corners left by the rotation, then a random horizontal flip is applied.

When applying these techniques the networks performance improved, with some fine tuning of variables I think acceptable results can be obtained.

The following were the best results, all had a batch size of 14 and trained for 25 epochs:

Test Loss: 1.040874	Test Loss: 0.849092	Test Loss: 0.849092
Test Accuracy of 0: 65% (13/20)	Test Accuracy of 0: 80% (16/20)	Test Accuracy of 0: 80% (16/20)
Test Accuracy of 1: 70% (14/20)	Test Accuracy of 1: 55% (11/20)	Test Accuracy of 1: 55% (11/20)
Test Accuracy of 2: 50% (10/20)	Test Accuracy of 2: 65% (13/20)	Test Accuracy of 2: 65% (13/20)
Test Accuracy of 3: 80% (16/20)	Test Accuracy of 3: 65% (13/20)	Test Accuracy of 3: 55% (11/20)
Test Accuracy of 4: 55% (11/20)	Test Accuracy of 4: 65% (13/20)	Test Accuracy of 4: 65% (13/20)
Test Accuracy of 5: 80% (16/20)	Test Accuracy of 5: 90% (18/20)	Test Accuracy of 5: 90% (18/20)
Test Accuracy (Overall): 66% (80/120)	Test Accuracy (Overall): 70% (84/120)	Test Accuracy (Overall): 68% (82/120)
Learning Rate: 0.00014	Learning Rate: 0.00015	Learning Rate: 0.00016

There appears to be a clear trend in the results, with the network performing much better when dealing with people from either extreme. This was something I hypothesized due to the most physically apparent ages being very young and very old, however towards the middle of the classes things are less transparent. It is clear that the network struggles to differentiate between the 'adult' years, which isn't surprising due to the volatility of peoples facial aging in that time, there are definitely cases within those classes where someone older than someone else appears facially younger.

Potential Creative Applications

A network such as this has many applications, neural networks in general are incredibly widespread in modern applications, however by focusing on facial classification there are avenues that other networks could not pursue.

One such example is the potential use of the network in security and surveillance. For example using this network you could search CCTV footage and the AI would flag anyone that appears to be a certain age, making finding people of interest in busy spaces far easier and quicker than combing through the crowds and footage manually. Another potential use in this area would be the automatic determination of the demographic attending events, cameras on the entrances would allow the AI to automatically detect and count the number of certain age ranges that attend events. This makes planning repeat events easier as hosts would know the attendees demographic. Unfortunately, security in particular is a very important area and the current accuracy of the network may not be feasible in such a context, as potentially costly mistakes would be commonplace with such results.

Whereas there is another area wherein this model would be very suitable, the mobile app industry. Apps like Snapchat and Instagram utilise filters on photos taken in real-time, many of these use neural networks and facial recognition, so this network could be used to create an age guessing filter for use in photo-messaging platforms.

Review

Overall, the network produced yielded results that were more than acceptable, having looked into other similar works I feel that 70% accuracy is very acceptable. The non-uniform nature of human faces, as well as the fickle unpredictable rules that govern the facial signs of age, achieving 70% accuracy is sufficient and I think given a larger dataset the results can be pushed higher.

Although, things could have been done differently throughout the development of the network. The dataset is an area that I feel could use improvement; despite large amounts of time spent manually cropping the images, there are still inconsistencies in the size and placement of the faces. Also, there are complications brought on by the use of celebrities in the dataset, for example the prevalence of editing and 'touching up' of images published in articles and magazines about famous actors / actresses and musicians. Collation of a new higher quality dataset and the further fine tuning of the model are the apparent next steps.

References

[1] Saha S., (2018) *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.* Available at: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (accessed: 16th January 2022)

[2] Tsang S., (2018) *Review: ResNeXt — 1st Runner Up in ILSVRC 2016 (Image Classification).* Available at: https://towardsdatascience.com/review-resnext-1st-runner-up-of-ilsvrc-2016-image-classification-15d7f17b42ac (accessed: 15th January 2022)

[3] Mustapha M.F. Mohamad N.M. Osman G, Hamid S.H., *(2021) Age Group Classification using Convolutional Neural Network (CNN)*. Available at: https://iopscience.iop.org/article/10.1088/1742-6596/2084/1/012028/pdf (accessed: 18th January 2022)

[4] Qawaqneh Z. Mallouh A.A. Barkana B.D., *Deep Convolutional Neural Network for Age Estimation based on VGG-Face Model.* Available at: https://arxiv.org/ftp/arxiv/papers/1709/1709.01664.pdf (accessed: 18th January 2022)

[5] Karras et al. (2019), *This Person Does Not Exist.* [website] Available at: https://thispersondoesnotexist.com/ (accessed: 14th January 2022)

[6]Ma E., (2020) *Enhancing ResNet to ResNeXt for image classification.* Available at: https://medium.com/dataseries/enhancing-resnet-to-resnext-for-image-classification-3449f62a774c#:~:text=Same%20as%20ResNet%2C%20having%20shortcuts,size%20for%20each%20individual%20block. (Accessed: 13th January 2022)

Bibliography

[1] PyTorch Team, *ResNext | PyTorch.* Available at: https://pytorch.org/hub/pytorch_vision_resnext/ (accessed: 14th January 2022)

[2] Xie S. Girshick R. Dollár P. Tu Z. & He K., *Aggregated Residual Transformations for Deep Neural Networks.* Available at: https://arxiv.org/abs/1611.05431 (Accessed: 12th January 2022)

[3] Chilamkurthy S., *Transfer Learning For Computer Vision Tutorial.* Available at: https://pytorch.org/tutorials/beginner/transfer_learning_tutorial.html (accessed: 14th January 2022)

[4] PyTorch Team, *TorchVision.models.* Available at: https://pytorch.org/vision/0.8/models.html#classification (accessed: 14th January 2022)

[5] Famous Birthdays [website] Available at: https://www.famousbirthdays.com/age/ (accessed 20th January 2022)