Facial Recognition using CNN and ResNet

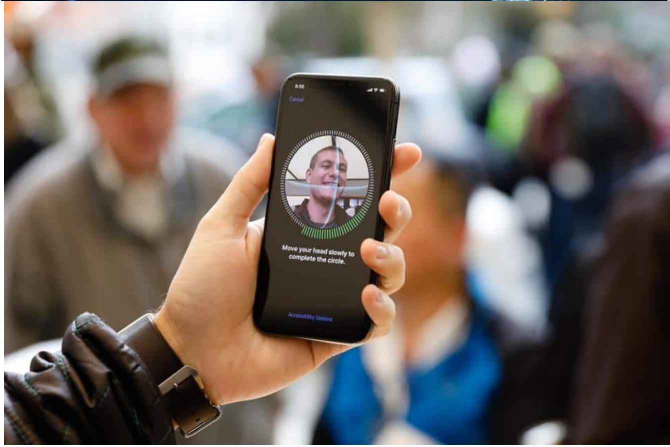
Group 6: Rosin Gu, Gustavo Nino, Mingzhan Yang, Shannon Ooi, Yixuan Deng, Yuchen Wang; Yaxin Yang

Introduction

Face recognition has become a popular technology in our daily lives nowadays.

For this project, we are interested in exploring the face recognition technology and the methodology behind it. We use two algorithms for this project, CNN (Convolutional Neural Network) and ResNet (Residual Neural Network).

We then compare the methodologies and accuracies of these two algorithms.



Data

We use a extensive data set from Kaggle of celebrity pictures:

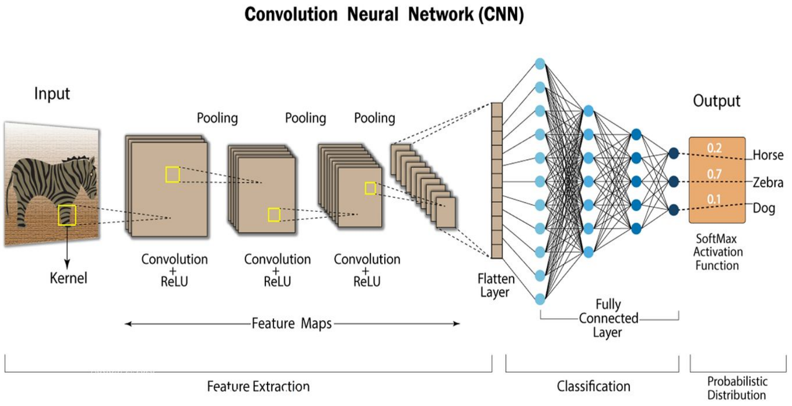
- 31 recognized hollywood celebrities with 82 images on average for each celebrity: **A total of 2,562 pictures**.
- **Images capture a wide range of scenarios:** various attire, accessories, and environmental contexts.
- We resize all images to a standard **size of 224x224 pixels**. We organize the dataset into **32 randomly selected batches**

Charlize Theron

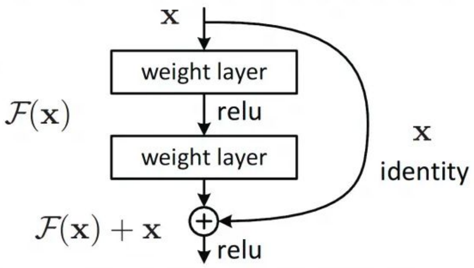
Brad Pitt

CNN

- Our CNN architectures and parameters are as follows:
  - 4 Convolution and max Pooling layers with Batch Normalization applied
  - The number of filters starts at 32, increases to 64 then drops down to 32 for the last convolution layer (32-64-64-32)
    - Our second model expands the number of neurons further to 96 before dropping down to (32-64-64-96-32)
  - 3x3 kernel size
  - ReLU activation
  - Dropout rate of 0.2
  - 1 hidden Dense Layer with ReLU activation and 128 units
  - Classification Dense Layer with softmax activation
  - Learning rate: 0.001, number of epoch: 20, 25



Residual Neural Network (ResNet)



- Motivation:
  - Increase layers without having the vanishing gradient problem
- ResNet50
  - 48 convolutional layers, 1 MaxPool layer, 1 average pool layer

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figures extracted from **"ResNet (34, 50, 101)…what actually it is ?", Chandini (2020)** https://medium.com/@aschandinip/resnet-34-50-101-what-actually-it-is-c63da24ba695

Result comparison of two model

- CNN
  - Structure: 32-64-64-96-32
  - Training accuracy is 93%
  - Validation accuracy is 40%
  - Epoch 50
  - time/epoch: 270s
- Resnet50
  - Training accuracy is 92%
  - Epoch 20
  - time/epoch: 540s
- Resnet101
  - Training accuracy is 99%
  - Epoch 20
  - Time/epoch: 890s

CNN - Difficulties and Parameter Tuning

- Difficulties :
  - Marginal training accuracy — only 93%
  - Poor Validation accuracy — only 40%
  - Long time — Different optimizers
  - Accuracy plateauing after epoch 20 — vanishing gradient
- Parameter Tuning:
  - Number of Layer
  - Optimizer
  - Epochs

ResNet - Difficulties and Parameter Tuning

- Difficulties：
  - Longer time（Compared with CNN)
  - Resnet 50/101/152 (Resnet 34-underfitting, Resnet 100-overfitting)
- Parameter Tuning:
  - Number of Layer
  - Epochs (easy to overfitting)

Conclusion

- Resnet outperform the CNN algorithm
  - 99% vs 93% in best scenario training model
- Resnet is computationally more intensive than CNN
- An exercise employing a validation set with CNN demonstrates an accuracy of 40%, indicating a potential issue of overfitting within our models.

Thank you～