Classification of Indians into North and South Indians

Under the Mentorship of
Prof. Suneeta Agarwal

GROUP : CS 23

- Tuhin Subhra Patra (20164142)
- Upamanyu Jamwal (20164169)
- Rajat Dipta Biswas (20164114)
- S Pranav Ganesh (20164098)

Introduction

- Humans are able to process a face in a variety of ways to categorize it by its identity, along with a number of other demographic characteristics, including ethnicity (or race), gender and age.

- Over the past few decades, a lot of effort has been devoted in the cognitive sciences areas, to discover how the human brain perceives, represents and remembers faces.

- Computational models have also been developed to gain some insight into this problem. Classification of people based on their ethnicity is one of the most challenging problems in today's world.

Motivations

- Accurate and swift classification of different races based on human face data in an uncontrolled environment is challenging especially in a multi-cultured country like India.

- Gleaned the power of the recent advances in computer vision and machine learning, we took the challenge to investigate whether or not it is possible to classify North and South Indians whose faces highly resemble each other.

- Previous researchers have worked on classifying Chinese, Korean and Japanese people based on their faces which led us to the idea of classifying Indians into North and South Indians.

Approach

- We intend to characterise the performance of computers on a fine-grained classification task.
- CNN architectures have been compared on the basis of coarse grained classification tasks like Cats vs Dogs, Dog Breed Classification in the recent past.
- We used CNN architectures to compare and classify people of north and south India. We take the challenge of comparing some state-of-the-art CNN architectures by using transfer learning.

Topics Covered

- Dataset Description
- Image Preprocessing
- Methods
- Results & Analysis
- Conclusion
- Future Work

Dataset Description

- We tried using Google Custom Search API
  - Requires manual filtering and editing
  - Majority were on makeup, studio images or memes.

- Very few facial images datasets labelled by Indian ethnicities in public domain.

- We used CNSIFD *(Centre for Neuroscience Indian Face Dataset)*
  - Heavily preprocessed data for custom use only
  - In MATLAB format

Dataset Description

Set 1

- It consisted of 459 face images collected with informed consent from volunteers in accordance with a protocol approved by the Institutional Human Ethics Committee of the Indian Institute of Science.

- Volunteers were photographed in high resolution (3648 x 2736 pixels) against a neutral background.

- Photographs were collected primarily from volunteers who declared that they as well as both parents belong to a north Indian or south Indian state.

Dataset Description

Set 2

- Set 2 consisted of 1,188 faces selected from the internet after careful validation.

- 128 typical first and 325 last names from each region were identified.

- Used Google Image search APIs to search for face photographs associated with combinations of these typical first and last names

Dataset Description

Testing set

- North - 125
- South - 125

Validation set

- North - 125
- South - 125

Training set

- North - 532
- South - 510

Table: **Gender Distribution of samples**

Region	Gender	Total
North	Male	317
	Female	465
South	Male	300
	Female	460

Figure: **Gender Distribution of samples**

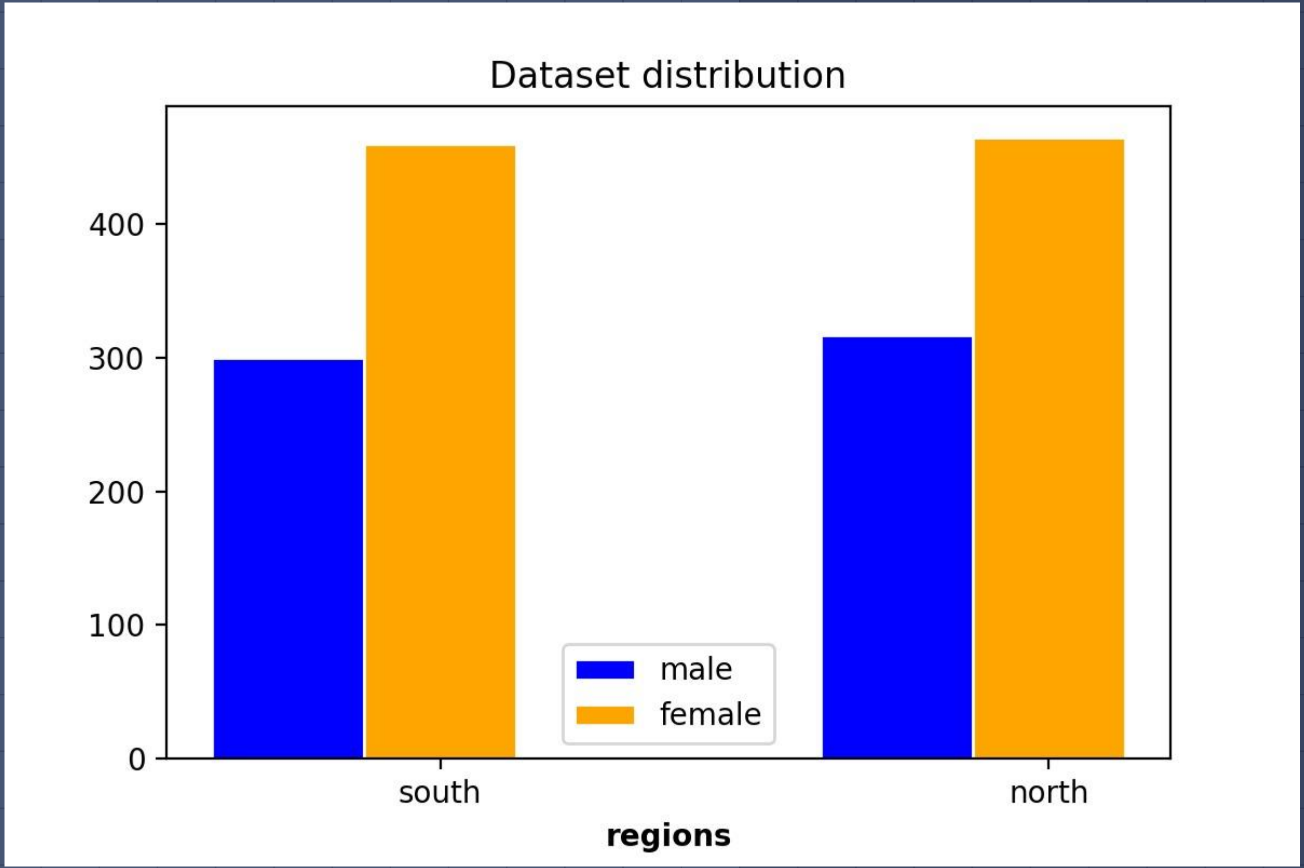


Figure: **Age distribution of samples**

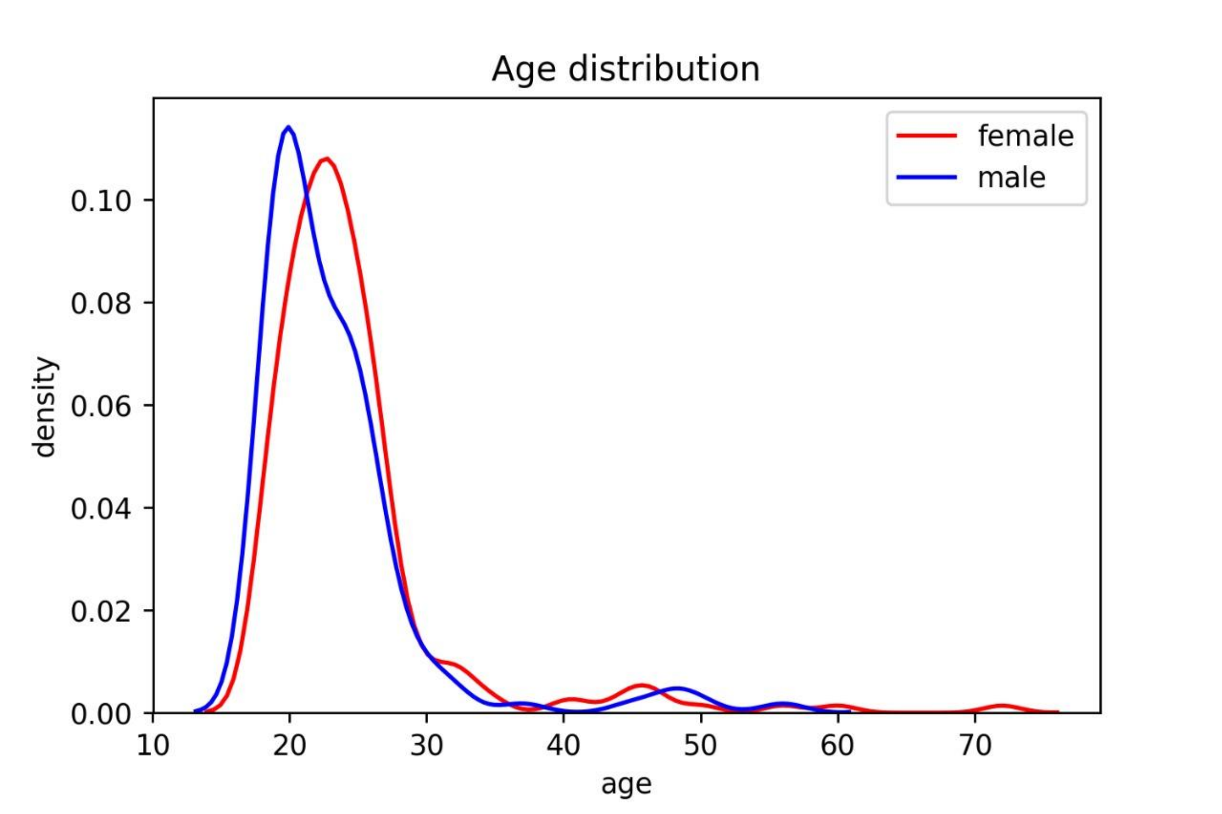


Image Preprocessing

STEP 1

The images were packed as MATLAB arrays. We extracted the grayscale images from the MATLAB format and converted it to CSVs and numpy arrays.

STEP 2

The images were cropped since each training data-point should have same dimension.

STEP 3

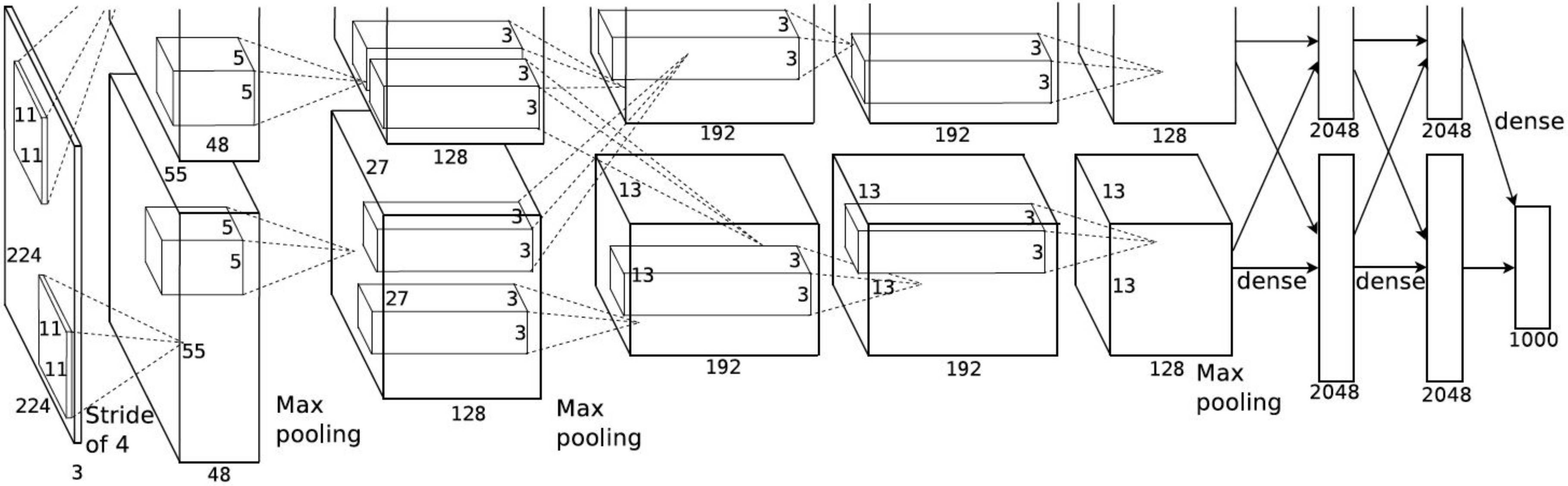
Since pretrained models in transfer learning are trained on colourful images only, we converted the grayscale image to PNG format which automatically assigned RGB values to the image.

Methods

- **AlexNet**

  It is a deep network composed of 5 convolutional layers followed by 3 fully connected layers. It uses ReLu (Rectified Linear Unit) for the non-linear part, instead of a Tanh or Sigmoid function which was the earlier standard for traditional neural networks. The advantage of the ReLu over sigmoid is that it trains much faster than the latter. ReLu layer is put after each and every convolutional and fully-connected layers (FC).

Figure: **AlexNet Architecture**



Picture source: http://web.ics.ei.tum.de/~karinne/Pdfs/MasterarbeitICS-FINALVERSION-Niklas.pdf

Methods

- **VGG**

VGGNet consists of 16 convolutional layers and is very appealing because of its very uniform architecture. Similar to AlexNet, only 3x3 convolutions, but lots of filters. The weight configuration of the VGGNet is publicly available and has been used in many other applications and challenges as a baseline feature extractor. However, VGGNet consists of 138 million parameters, which can be a bit challenging to handle.

Figure: **VGG Architecture**

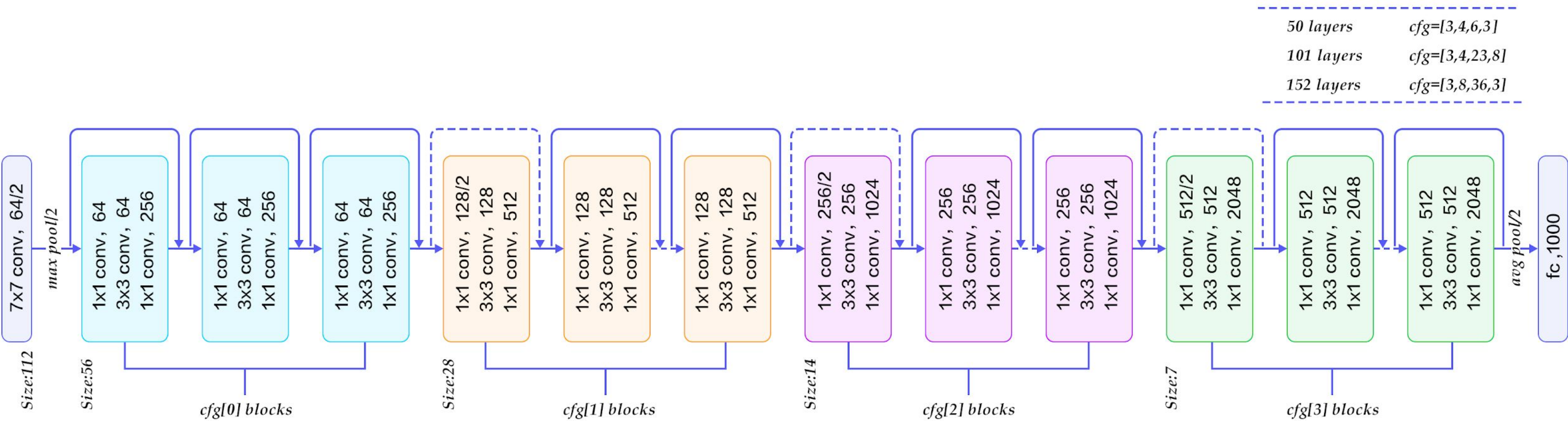
Picture source: https://www.sciencepubco.com/index.php/ijet/article/download/18588/8470

Methods

- **ResNet**

  With network depth increasing, degradation problem, optimizing difficulty and vanishing/exploding gradients reduces efficiency. Resnet architecture came up with the idea of skip connections with the hypothesis that the deeper layers should be able to learn something as equal as shallower layers. The solution is copying the activations from shallower layers and setting additional layers to identity mapping.These connections are enabled by skip connections. The architecture is similar to the VGGNet consisting mostly of 3X3 filters.

Figure: **ResNet Architecture**



Picture source: http://www.cse.ohio-state.edu/~panda/5194/slides/3.a-3.b.caffe_caffe2.pdf

Results & Analysis

Figure: **Training vs Validation Set Loss**

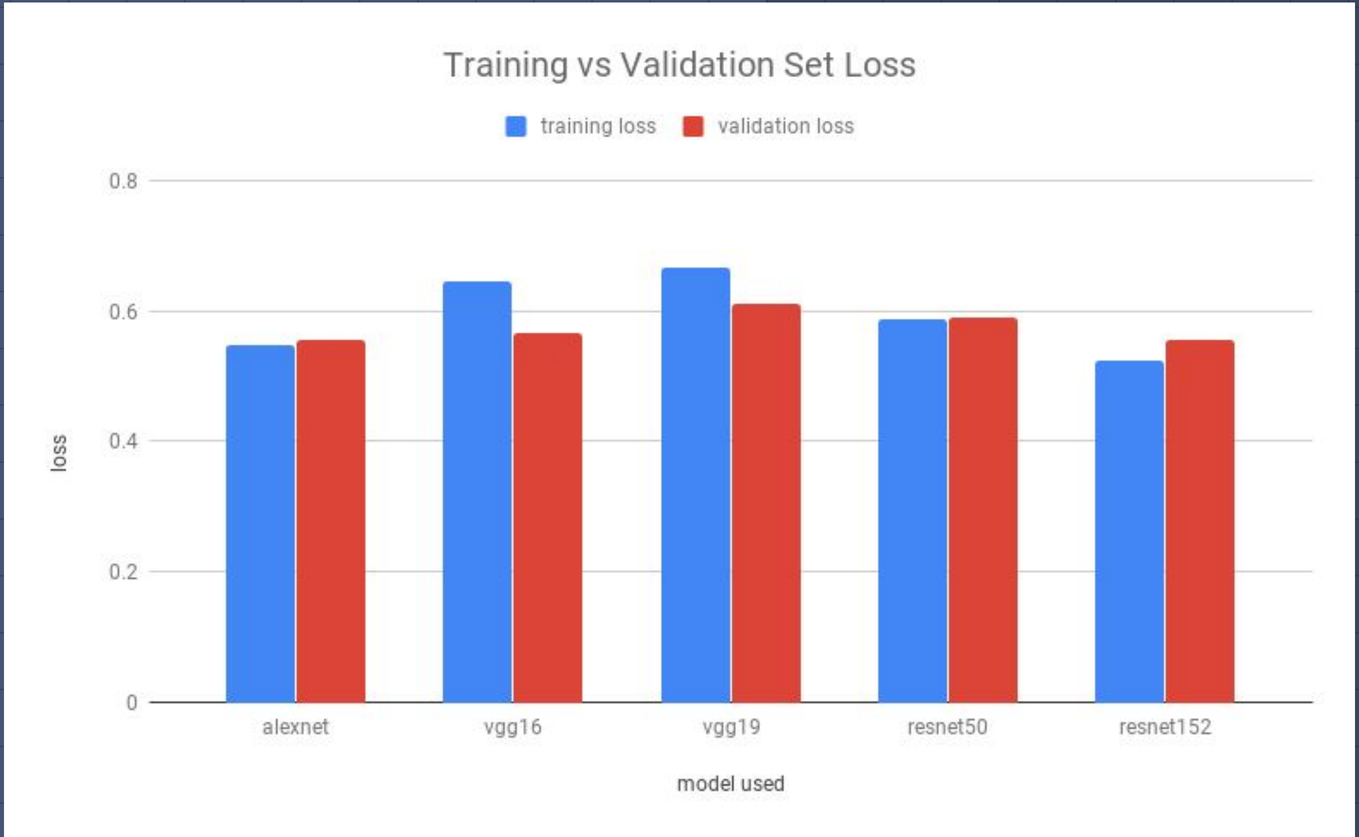


Figure: **Accuracy after 12 epochs**

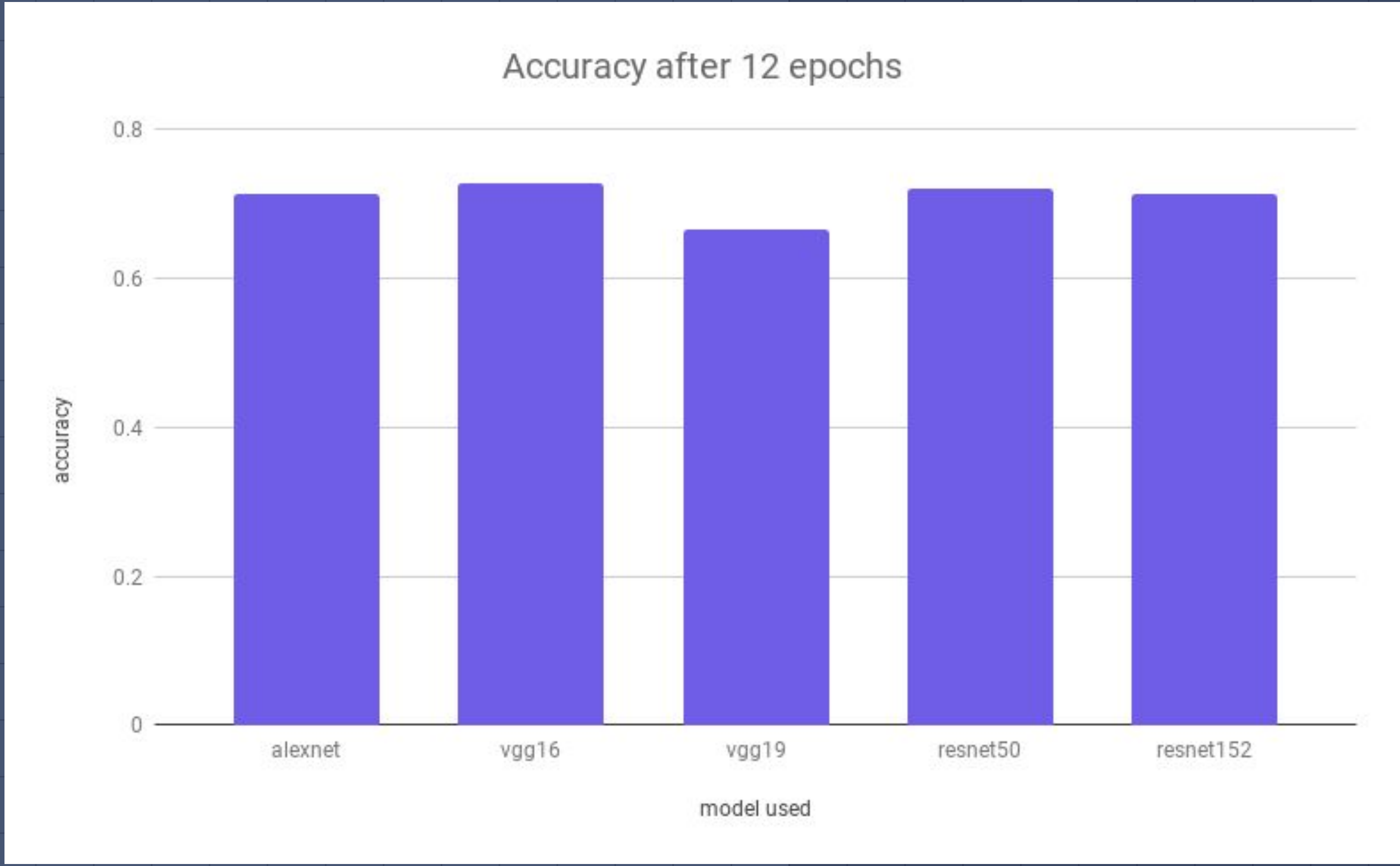
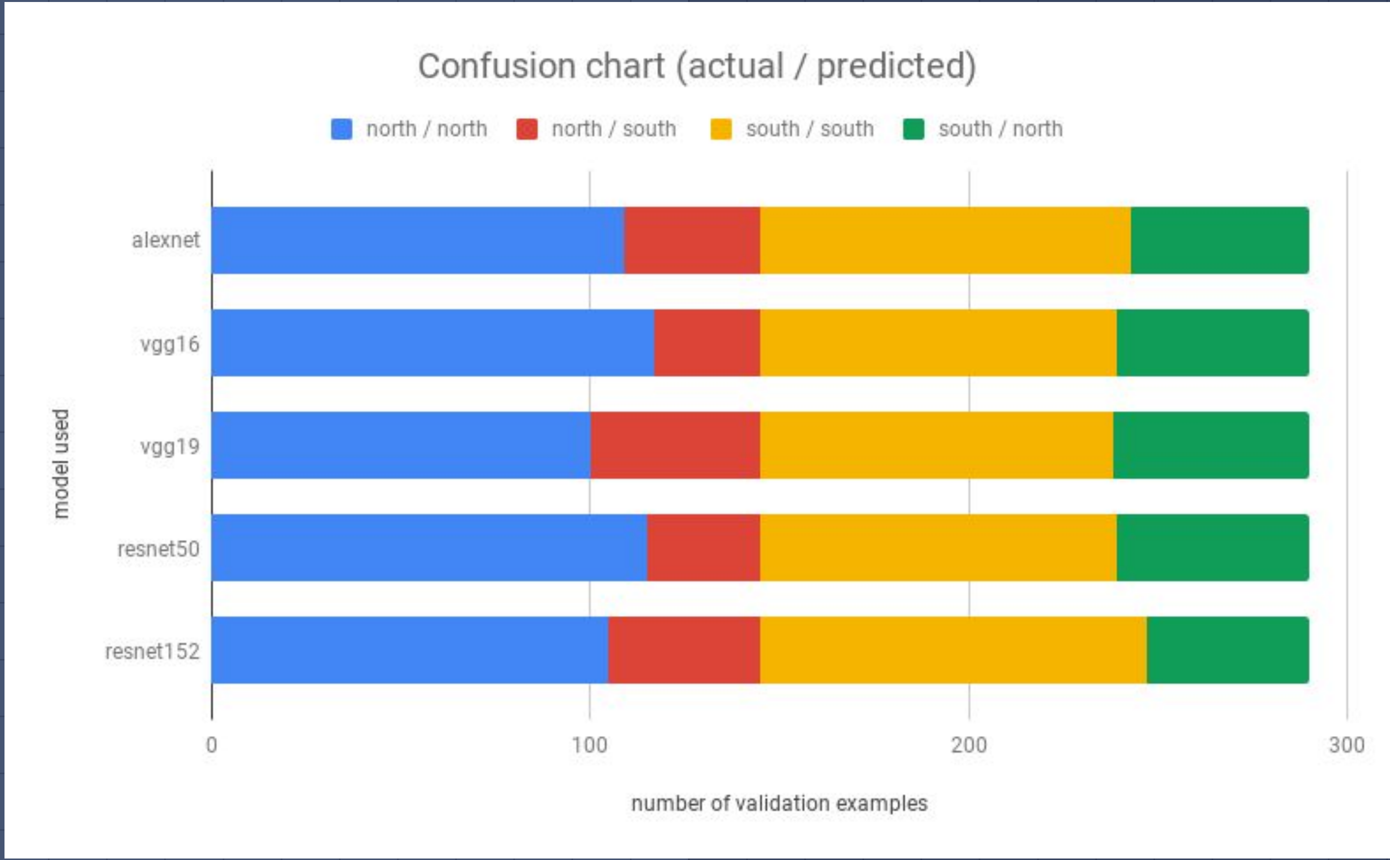


Figure: **Confusion Chart (Actual vs Predicted Labels)**



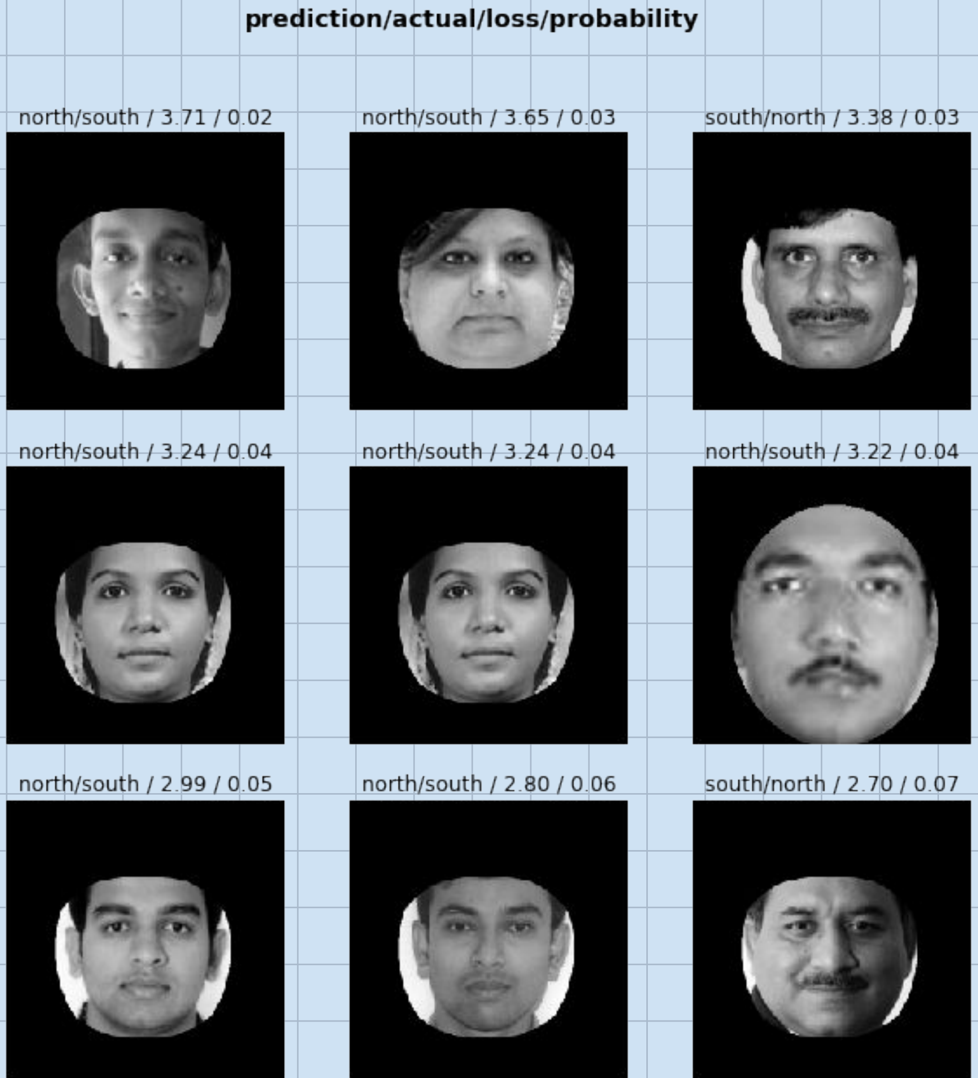


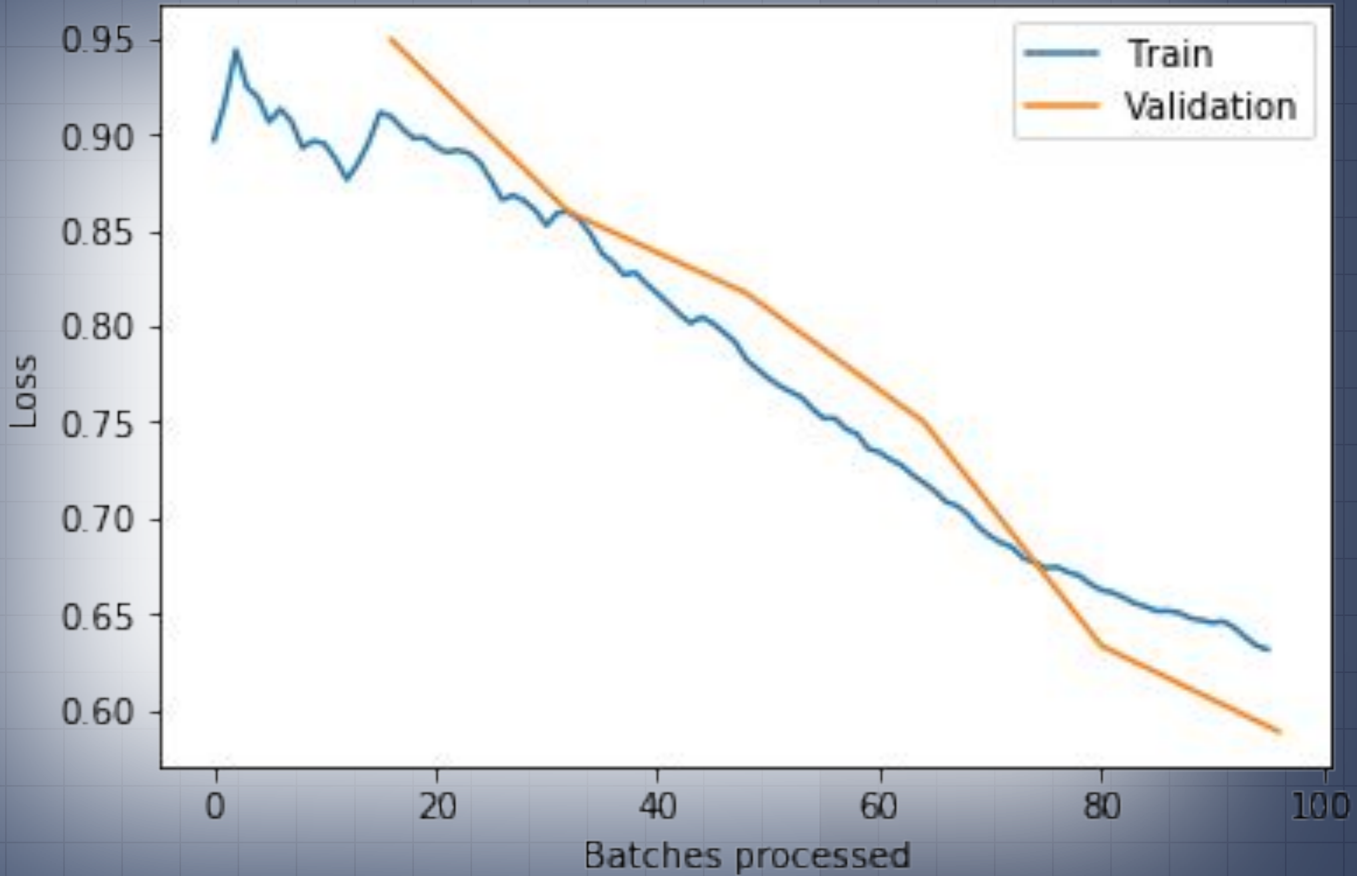
Figure: **Top Losses for ResNet50**

```
[105] learn.fit_one_cycle(12, max_lr=slice(1e-5,1e-4))
```

epoch	train_loss	valid_loss	accuracy	time
0	0.779558	0.667272	0.593103	00:27
1	0.794279	0.643220	0.641379	00:27
2	0.802565	0.612118	0.700000	00:28
3	0.765868	0.610731	0.679310	00:27
4	0.731630	0.602753	0.693103	00:26
5	0.698876	0.621028	0.682759	00:27
6	0.681923	0.612765	0.710345	00:26
7	0.646857	0.611278	0.717241	00:28
8	0.612747	0.601574	0.727586	00:27
9	0.589422	0.584658	0.724138	00:26
10	0.587082	0.588168	0.724138	00:26
11	0.565178	0.591550	0.720690	00:27

Figure: **Training Summary for ResNet50 (Learning Rate manually distributed across layers)**

Figure: **Loss vs Batches Processed for ResNet152 for 6 epochs**



Conclusion

- We have characterized machine performance on a hard race classification task using VGG, ResNets and AlexNets.

- Our main finding is that many computational models can achieve human levels of performance (64%) or even better.

- Assuming that the kind of faces in our dataset are similar to the training examples experienced by humans - this raises the interesting question of what features humans extract from faces, and how they learn it

Future Works

- Fine grained race classification can be improved with more advanced architectures that permit us to use and build more deeper neural nets.
- Improved classifications can help us reveal the extent to which our face structures depend on our geolocations.
- Extending this study to finer grained classification can reveal how much intermingling of facial features is present as a function of distance.

Future Works

- Studying finer-grained race classifications can greatly improve coarse grained classifications such as Asians, Africans, Hispanics, etc.

- Can these classifications which help us in revealing similarities between ethnic groups, give us more clues to the spatial evolution of ethnic groups over time?

THANK YOU