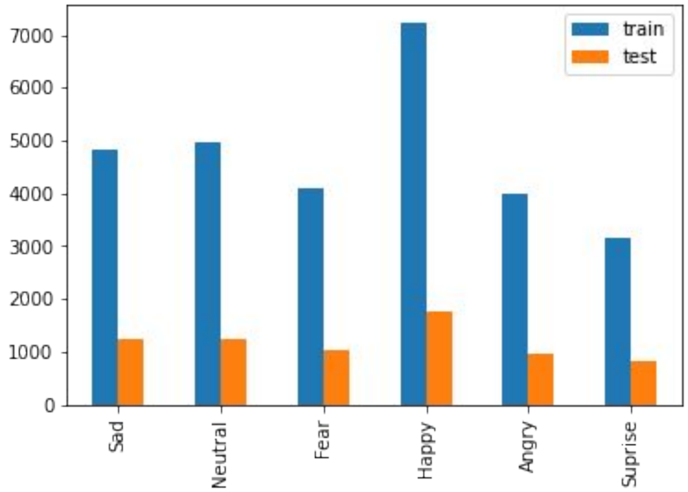
Emotion Recognition in Still Images

Team Members:

- Hanzallah Azim Burney
- Sayed Abdullah Qutb
- Balaj Saleem
- Mehmet Alper Genç
- Burak Mutlu

Facial Recognition Dataset

- “Facial Recognition Dataset” supplied by Gotam Dahiya on Kaggle
- Emotions include happiness, sadness, anger, fear, neutral, and surprise
- Training set contains 28,079 image samples
- Testing set includes 7,178 sample images
- Each image is a 48x48 grayscale image



ResNet50V2 and ResNet101V2

- Construct is based on pyramidal cells in the cerebral cortex [4]
- Residual building blocks allow forward and backward signals to be directly propagated from one block to any other block [2]
- Uses identity mappings as the skip connections [2]
- Solves the vanishing gradient problem that deep CNN architectures face [3]
- Reuse the activations generated by the previous layers until the adjacent layer has managed to learn its weight [4]

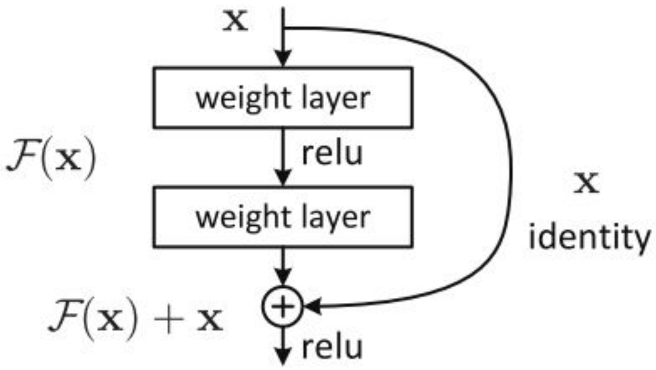


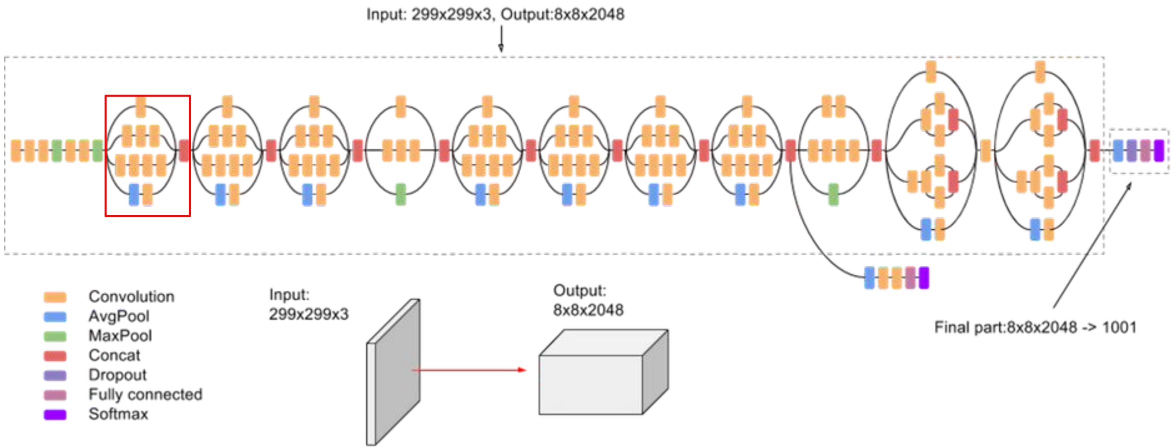
Fig 1. Residual Models [3]

	Top-1 Accuracy	Top-5 Accuracy
ResNet50V2	0.749	0.921
ResNet101V2	0.746	0.928

Fig 2. ResNet model from Keras [1]

InceptionV3 and InceptionResNetV2

- InceptionV3: “Greater than 78.1% accuracy on the ImageNet dataset” [5]
- Made up of numerous building blocks with different functions [5]
- Convolutions are factorized through grid size reduction and filter concatenation [6]



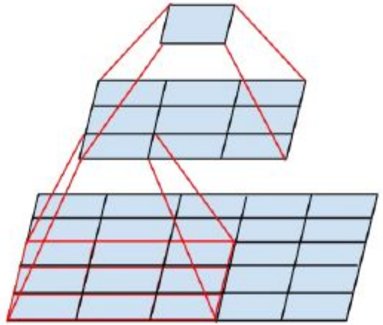
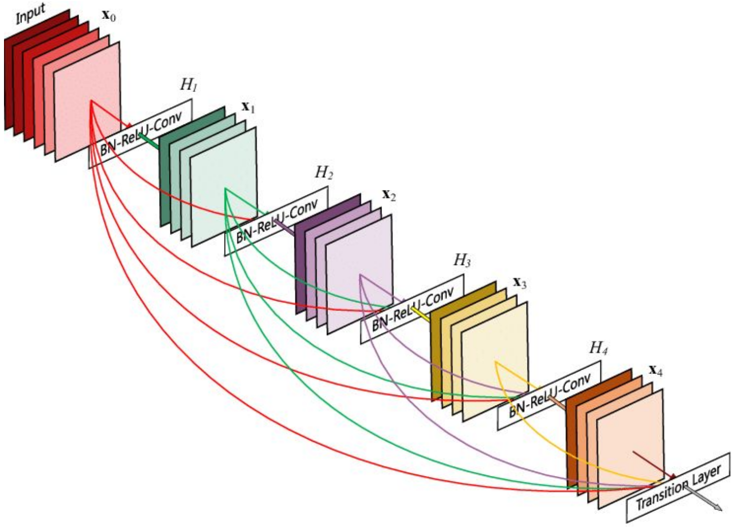


Figure 1. Mini-network replacing the 5×5 convolutions.

DenseNet Architecture

- DenseNet refers to densely connected CNNs
- A layer passes its input feature-maps to all subsequent layers; full layer-connection
- The layer at the level k has k input feature-maps
- BN - ReLU - Conv(1x1) - BN - ReLU - Conv(3x3)

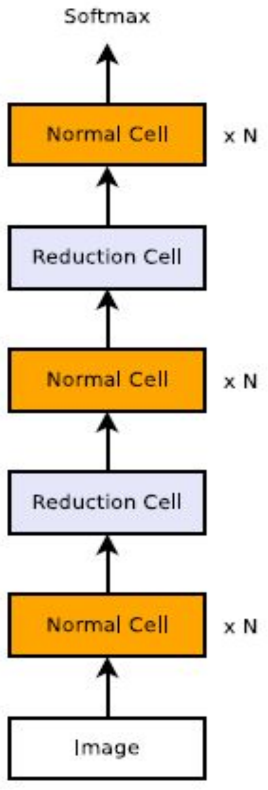


DenseNet121 - Accuracy: (0.750, 0.923)

DenseNet169 - Accuracy: (0.762, 0.932)

NasNet

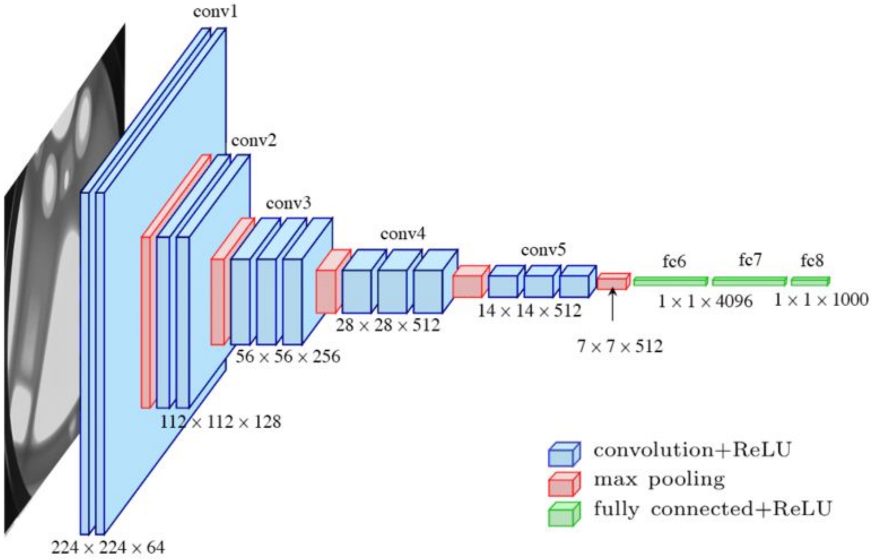
- Neural Architecture Search Network (Convolutional Neural Network)
- Large datasets, 1000+ categories
- Search for the best convolutional layer on CIFAR10, then apply on ImageNet
- Blocks found through reinforcement search method
- NasNet Large: 331x331 pixels, more parameters, approx. 11 million. 82.5%
- NasNet Mobile: Lighter, compact, mobile-platform, 224x224 pixels, less parameters approx. 5 million. 74.4%



CIFAR10 Architecture

VGG

- State of the art Image Classification model
- Uses Convolutional Neural Networks (CNNs)
- 16-19 Layers
- Very small 3x3 convolution filters
- Pre-trained on Image-net



Preprocessing

- ‘On the fly’ image preprocessing and conversion to a pixel feature set using keras

- Mean-centering, normalization, and standardization of images.

- ZCA whitening
  - Similar to dimensionality reduction using PCA
  - Retains the same spatial arrangement of pixels
  - It removes redundant pixels through a linear algebra based operation

Evaluation Results - Epochs and Layers

- Increasing layers via using different models
- With more layers, the **model overfits** such as InceptionResNetV2 (162 layers)
- With very few layers, the **model underfits** such as VGG16 (16 layers)
- In one epoch network **sees the whole data**
- Increasing epoch allows **weight optimisation** using optimizer
- Increasing epochs can lead to **overfitting** but our models seem to avoid it

	10	15
ResNet50V2	0.477	0.535
ResNet101V2	0.479	0.509
InceptionV3	0.462	0.538
InceptionResNetV2	0.403	0.514
DenseNet121	0.502	0.534
DenseNet169	0.519	0.516
VGG16	0.251	0.251
VGG19	0.251	0.251
NasNetMobile	0.251	0.239

Accuracy Table for epochs

Evaluation Results - Learning Rate

- Learning rate is a hyper parameter that determines the extent the weight of the model change upon each iteration.
- Following were observed:
  - Higher learning rate meant higher learning time i.e. model took longer to converge
  - Higher learning rate gave lower (or equal in rare cases) accuracies since sub optimal solution was reached
  - Keras library dynamically modifies learning rate as model trains
- Below is an example of the effect of learning rate on Inception models

Model	Learning Rate	Loss	Accuracy
InceptionV3	1	1.906	0.200
InceptionResNetV2	1	1.841	0.251

Model	Learning Rate	Loss	Accuracy
InceptionV3	0.01	1.205	0.524
InceptionResNetV2	0.01	1.482	0.403

Evaluation Results - Optimizers

- Used to minimize the loss function
- **RMSprop**: exponentially weighted average of the squares of past gradients
- **SGD**: Sample batch from whole of dataset at each iteration.
- **Adam:** Combines 2 other optimizers (Momentum, RMSprop)
- RMSProp performed more accurate (due to Recurrent Neural Networks)

	RMSprop	SGD	Adam
ResNet50V2	0.411	0.440	0.624
ResNet101V2	0.479	0.466	0.453
InceptionV3	0.462	0.472	0.416
InceptionResNetV2	0.403	0.527	0.288
DenseNet121	0.502	0.444	0.444
DenseNet169	0.519	0.438	0.458
VGG16	0.251	0.251	0.251
VGG19	0.251	0.251	0.251
NasNetMobile	0.251	0.177	0.250

Accuracy Table for optimizers

Evaluation Results - Activation Functions

- Used to **rank the features** for prediction
- Activations with **smoother gradients** get better results
- Only function to get desirable results for all models is **softmax**
- **Best-performing** function for all models is softmax
- Softmax is best choice for an **image classification task**

	Softmax	Relu	Sigmoid
ResNet50V2	0.504	0.219	0.135
ResNet101V2	0.380	0.343	0.135
InceptionV3	0.517	0.426	0.136
InceptionResNetV2	0.471	0.275	0.264
DenseNet121	0.531	0.404	0.457
DenseNet169	0.497	0.419	0.486

Accuracy Table for activation functions

Evaluation Results - Normalization

- For VGG16 and InceptionV3, feature-wise normalization increased the accuracy
- ResNet50V2, ResNet101V2, InceptionResNetV2 and DenseNet models resulted in lower accuracies when data was feature-wise normalized

Model	Normalization	Loss	Accuracy
ResNet50V2	True	1.639	0.411
ResNet101V2	True	1.346	0.479
InceptionResNet V2	True	1.482	0.403
DenseNet121	True	1.262	0.502
DenseNet169	True	1.299	0.519

Model	Normalization	Loss	Accuracy
ResNet50V2	False	1.475	0.433
ResNet101V2	False	1.302	0.497
InceptionResNet V2	False	4.707	0.459
DenseNet121	False	1.245	0.525
DenseNet169	False	1.853	0.525

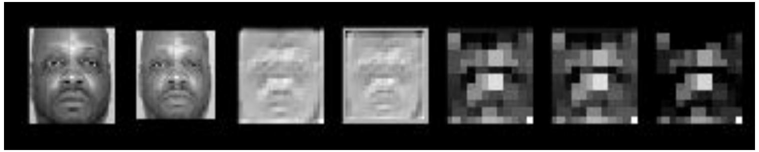
- VGG19 and NasNetMobile models had differences smaller than 10^{-3} in accuracies

Model	Normalization	Loss	Accuracy
VGG 19	True	1.760	0.251
NasNetMobile	True	1.176	0.251

Model	Normalization	Loss	Accuracy
VGG 19	False	1.758	0.251
NasNetMobile	False	1.89	0.251

Model Visualizations

ResNet50V2

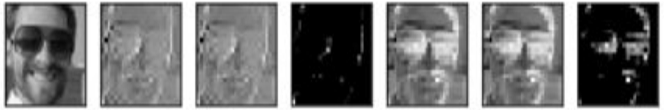


InceptionV3

DenseNet121

VGG16

NasNet Mobile



Conclusions

- Probability of random guessing was ~16.67%
- Deep Learning models generally performed above that
- Layers between 100 and 200 worked the best
- Epoch 15, learning rate = 0.01, SGD optimization, Softmax activation
- Emotion recognition is a viable problem to be solved using deep learning

References

[1] Team, K., 2020. *Keras Documentation: Keras Applications*. [online] Keras.io. Available at: <https://keras.io/api/applications/> [Accessed 15 November 2020].

[2] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Identity Mappings in Deep Residual Networks. *Computer Vision – ECCV 2016*, pp.630-645.

[3] Fung, V., 2020. *An Overview Of Resnet And Its Variants*. [online] Medium. Available at: <https://towardsdatascience.com/an-overview-of-resnet-and-its-variants-5281e2f56035> [Accessed 15 November 2020].

[4] En.wikipedia.org. 2020. *Residual Neural Network*. [online] Available at: <https://en.wikipedia.org/wiki/Residual_neural_network> [Accessed 15 November 2020].

[5] "Advanced Guide to Inception v3 on Cloud TPU | Google Cloud", Google Cloud, 2020. [Online]. Available: https://cloud.google.com/tpu/docs/inception-v3-advanced. [Accessed: 15- Nov- 2020].

[6] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision", arXiv.org, 2015. [Online]. Available: https://arxiv.org/abs/1512.00567. [Accessed: 15- Nov- 2020].

[7] Huang, G., Liu, Z., van der Maaten, L. and Weinberger, K., 2016. Densely Connected Convolutional Networks. [online] arXiv.org. Available at: <https://arxiv.org/abs/1608.06993> [Accessed 16 November 2020].