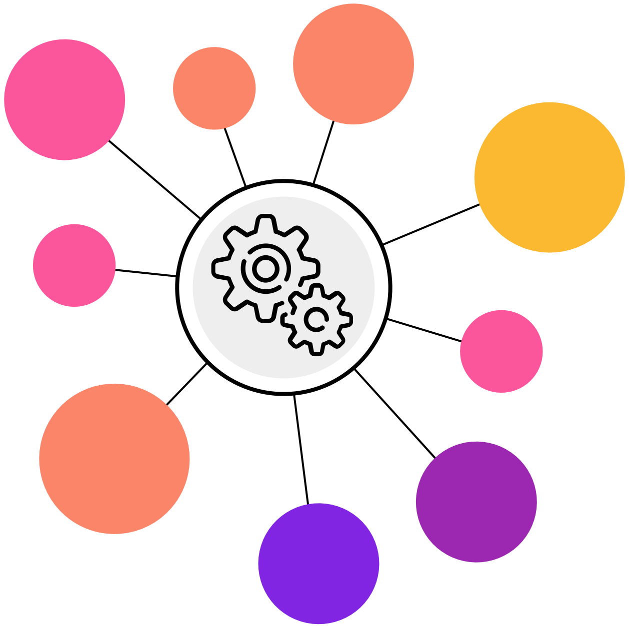


# Capstone Project

## Deep Learning Methods for Facial Emotion Recognition

## Transfer Learning Optimization

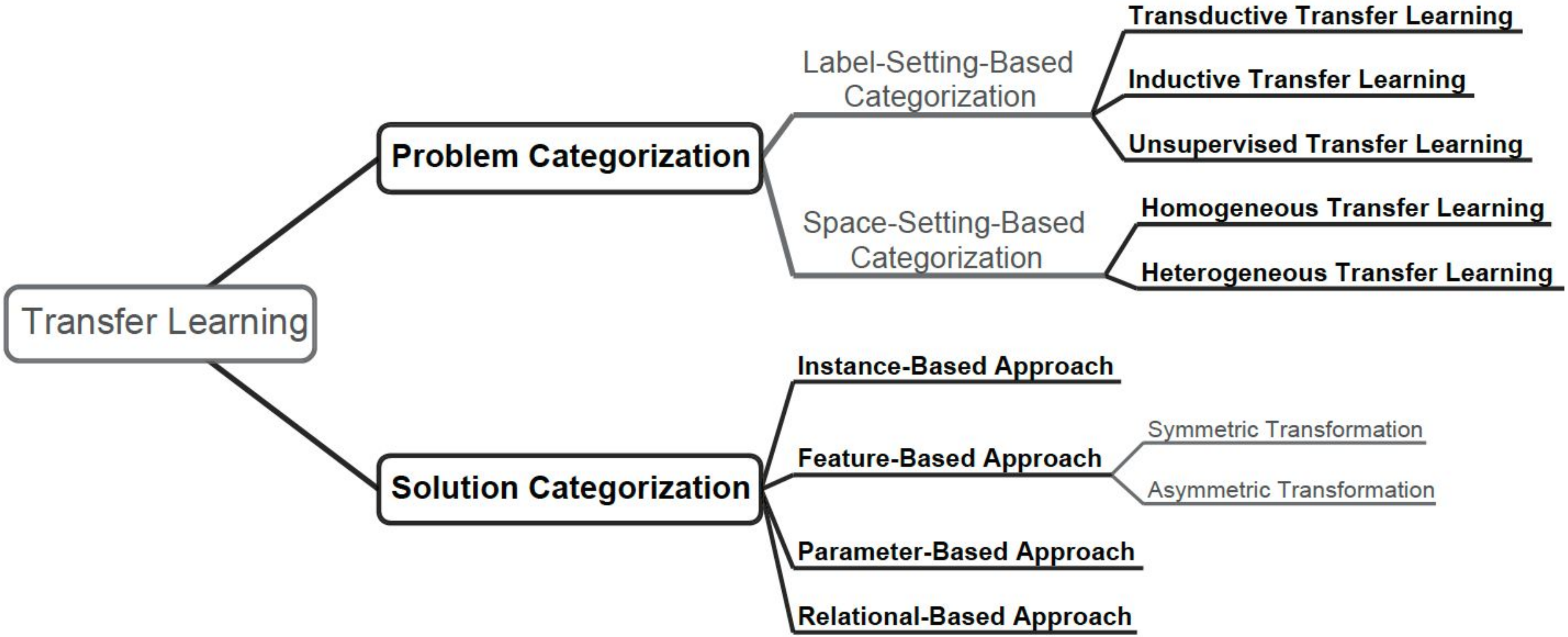
Monica Palacios Boyce, Ph.D.

Link to original capstone presentation

# These slides include the original presentation in the appendix

Link to original capstone presentation

# Transfer Learning is a large topic




Zhuang, F., et al (2020) A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1) 43-76. [link]

# Problem Definition

Recognizing accurate emotions in facial images can provide a deeper understanding of the user and situation in which the image was obtained.

Convolutional Neural Network models (CNNs) have been developed to process image data to learn higher order patterns (features) that can yield predictions of value on new images.

---

**In the first version of this capstone project, the use of transfer learning models as an alternative to the custom CNN model did not yield improved performance.**

---

**This current project aims to explore the proper design and use of these pre-trained CNN models on the FER 2013 dataset.**

Three pre-trained CNN models are evaluated:

- VGG16
- ResNet101
- EfficientNet B2

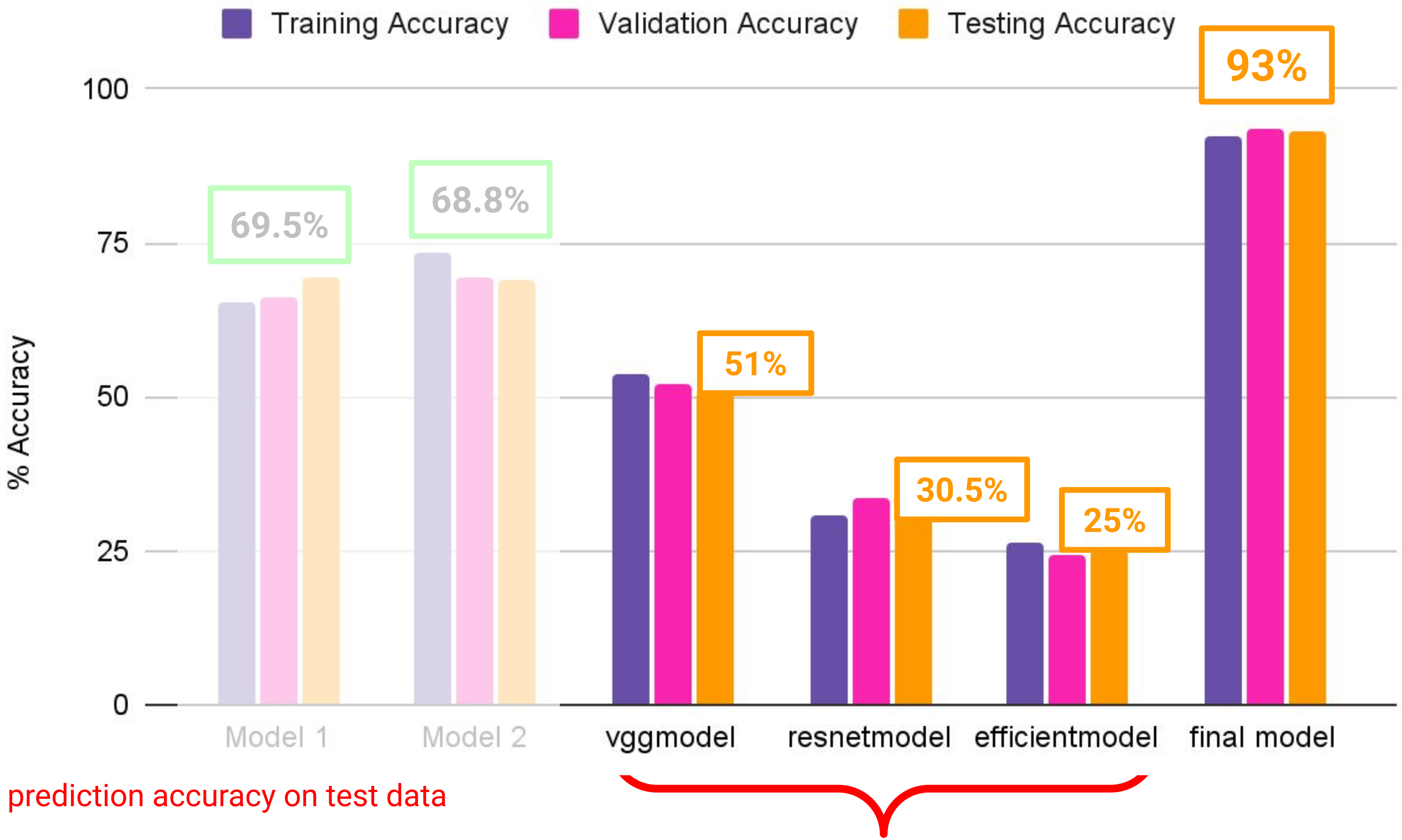
---

**Part 1:** Instead of using a single pre-trained layer from selected CNNs (as in the original capstone project), the entire frozen feature - extraction layers (the convolutional blocks) of the CNN models are trained on the FER 2013.

**Part 2:** The feature - extraction layers (the convolutional blocks) of the models are **UNFROZEN** and then trained on the FER 2013.

# Previous Transfer Learning Performance

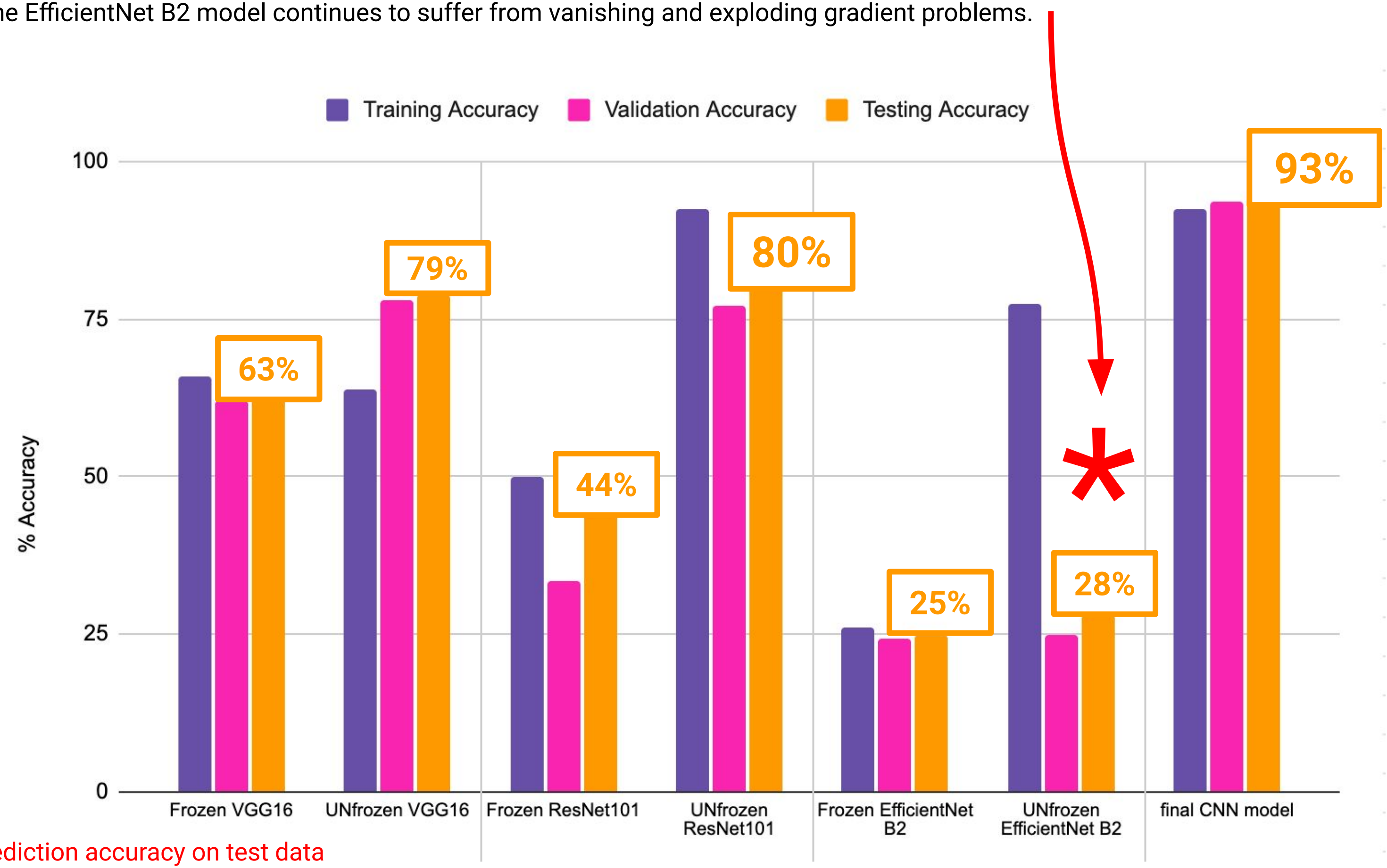
As seen in the chart below, the three transfer learning models significantly underperformed in comparison to the less complex CNN models built during the capstone project.




% indicates prediction accuracy on test data

# Improved Transfer Learning Performance

Changes in the handling of the pre-trained models had a positive impact on model performance for the VGG16 and ResNet101 models. The EfficientNet B2 model continues to suffer from vanishing and exploding gradient problems.




% indicates prediction accuracy on test data

# Summary of NEW findings - data table

| Model | Training Accuracy | Validation Accuracy | Testing Accuracy |
| --- | --- | --- | --- |
| Frozen VGG16 + new FC | 65.8% (0.6583) | 62% (0.6205) | 62.5% (0.6250) |
| UN-frozen VGG16 + new FC | 64% (0.6367) | 78% (0.7758) | **79%** (0.7891) |
| Frozen ResNet101 model + new FC | 50% (0.4959) | 33.5% (0.3351) | 44% (0.4375) |
| UN-frozen ResNet101 model + new FC | 92.4% (0.9240) | 77% (0.7655) | **80%** (0.7969) |
| Frozen EfficientNet B2 model + new FC | 26% (0.2621) | 24.4% (0.2443) | 25% (0.25) |
| UN-frozen EfficientNet B2 model + new FC | 77.4% (0.7742) | 25% (0.2489) | 28% (0.2812) |
| **Final custom CNN Model - RGB** | 92.4% (0.9243) | 93.6% (0.9356) | **93.3%** (0.9331) |

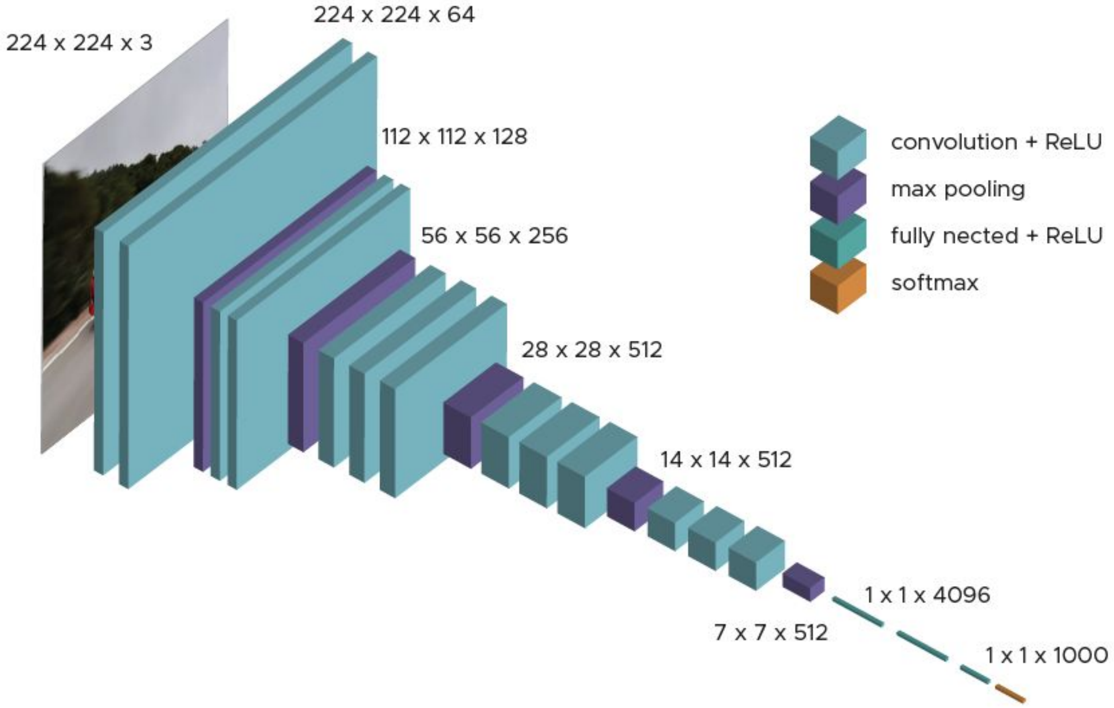
FC = fully connected layer

# VGG16

# VGG16 Architecture

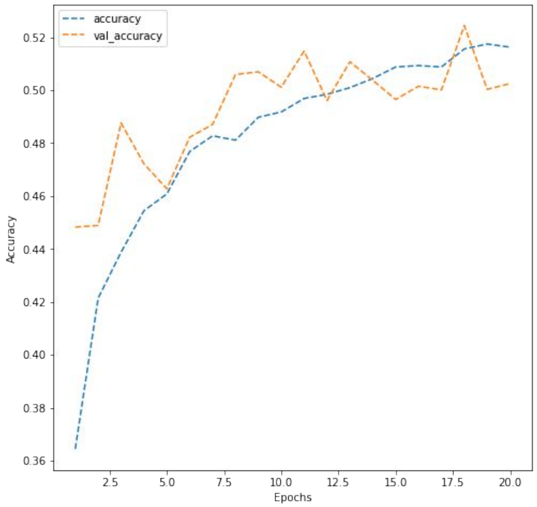
**16 layers of VGG16**

1.Convolution using 64 filters
2.Convolution using 64 filters + Max pooling
3.Convolution using 128 filters
4. Convolution using 128 filters + Max pooling
5. Convolution using 256 filters
6. Convolution using 256 filters
7. Convolution using 256 filters + Max pooling
8. Convolution using 512 filters
9. Convolution using 512 filters
10. Convolution using 512 filters+Max pooling
11. Convolution using 512 filters
12. Convolution using 512 filters
13. Convolution using 512 filters+Max pooling
14. Fully connected with 4096 nodes
15. Fully connected with 4096 nodes
16. Output layer with Softmax activation with 1000 nodes.

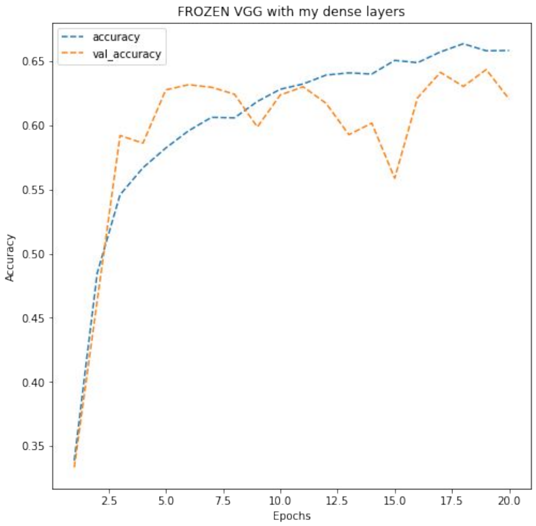


| Model | Training Accuracy | Validation Accuracy | Testing Accuracy |
|---|---|---|---|
| Final layer of VGG16 + new FC | 53.8% | 52% | 51% |
| Frozen VGG16 + new FC | 65.8% | 62% | 62.5% |
| UN-frozen VGG16 + new FC | 64% | 78% | 79% |

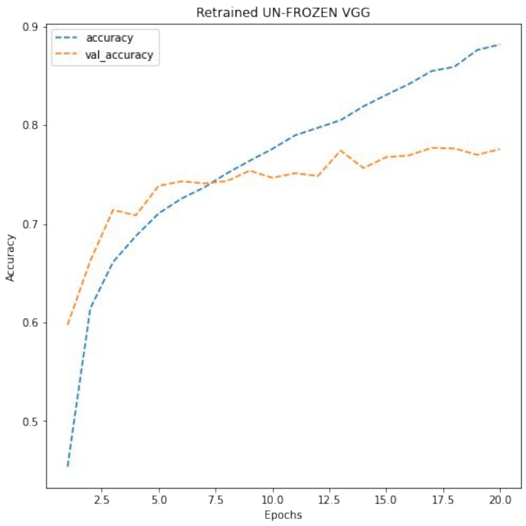
FC = fully connected layer




Final Layer VGG




FROZEN




UN-FROZEN

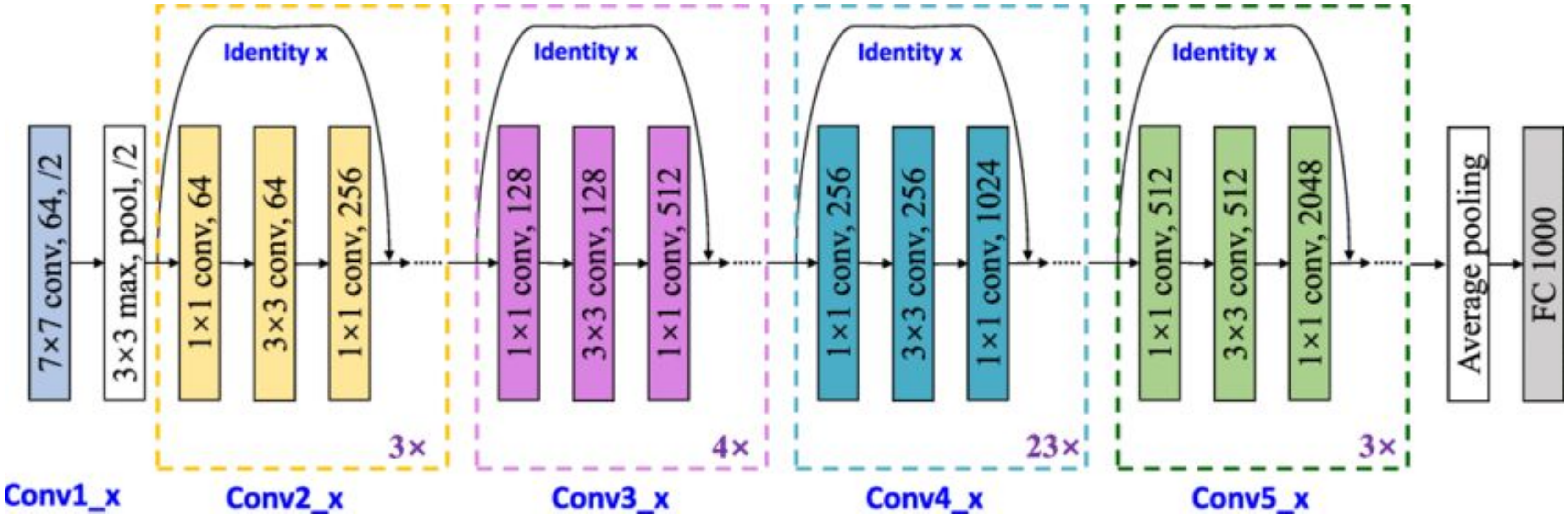
# Key Findings - VGG16

The FROZEN VGG16 (using all of the pre-trained feature-extraction layers with unchanging weights) provided a modest improvement over the attenuated VGG16 model discussed in the original capstone project.

The UNFROZEN VGG16 (using all of the feature-extraction layers with new weights trained on the FER 2013 dataset) provided a significant improvement in predictive performance (79% vs 51%)

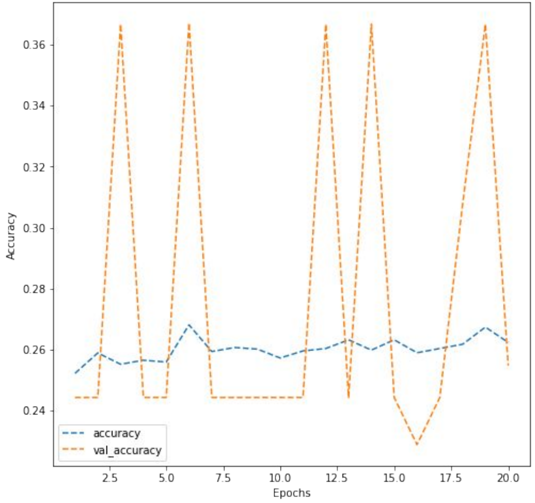
# ResNet101

# ResNet101 Architecture

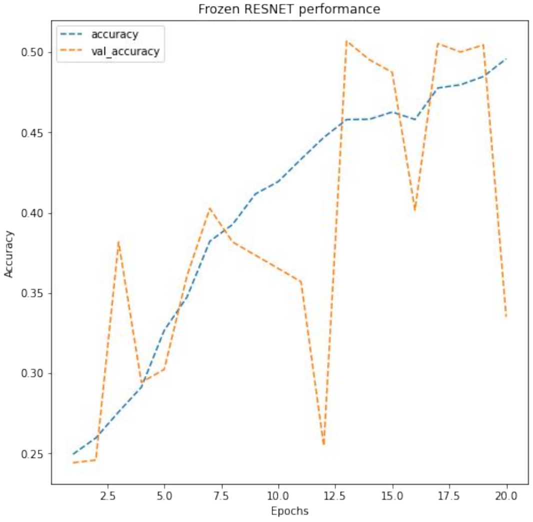


| Model | Training Accuracy | Validation Accuracy | Testing Accuracy |
|---|---|---|---|
| Final layer of Resnet + new FC | 30.6% | 33.4% | 30.5% |
| Frozen ResNet101 model + new FC | 50% | 33.5% | 44% |
| UN-frozen ResNet101 model + new FC | 92.4% | 77% | 80% |

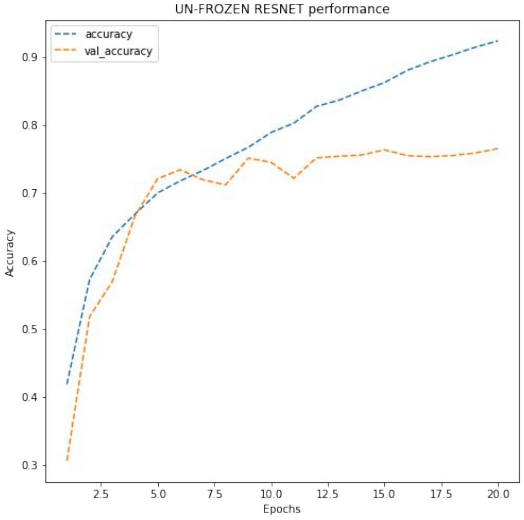
FC = fully connected layer




Final Layer resnet




FROZEN




UN-FROZEN

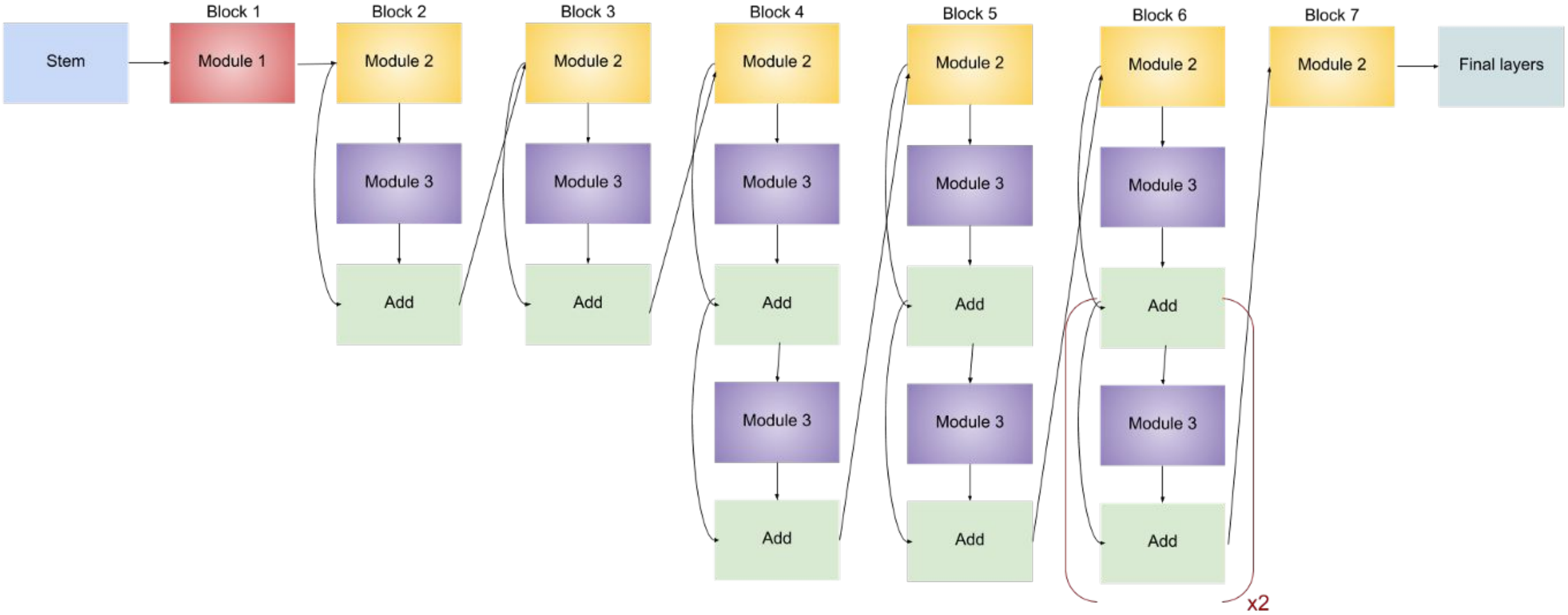
# Key Findings - ResNet101

The FROZEN ResNet101 (using all of the pre-trained feature-extraction layers with unchanging weights) provided a modest improvement over the attenuated resnet model in the original capstone project.

The UNFROZEN ResNet101 (using all of the feature-extraction layers with new weights trained on the FER 2013 dataset) provided a significant improvement in predictive performance (80% vs 30.5%)

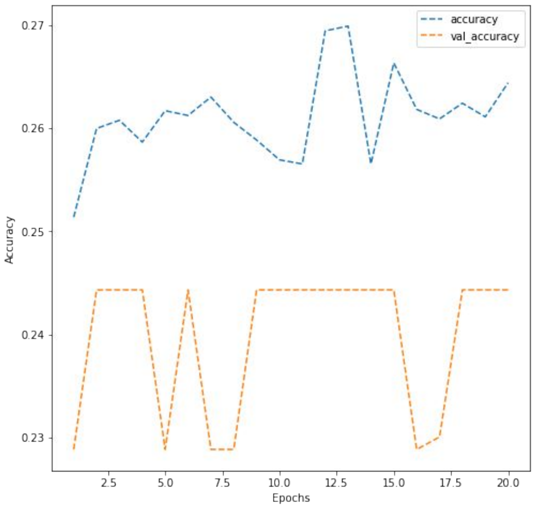
# EfficientNet B2

# EfficientNet B2 Architecture

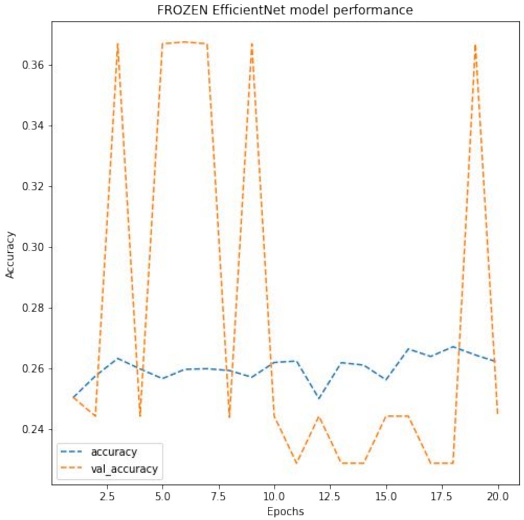


| Model | Training Accuracy | Validation Accuracy | Testing Accuracy |
| --- | --- | --- | --- |
| Final layer of EfficientNet + new FC | 26.3% | 24.4% | 25% |
| Frozen EfficientNet B2 model + new FC | 26% | 24.4% | 25% |
| UN-frozen EfficientNet B2 model + new FC | 77.4% | 25% | 28% |

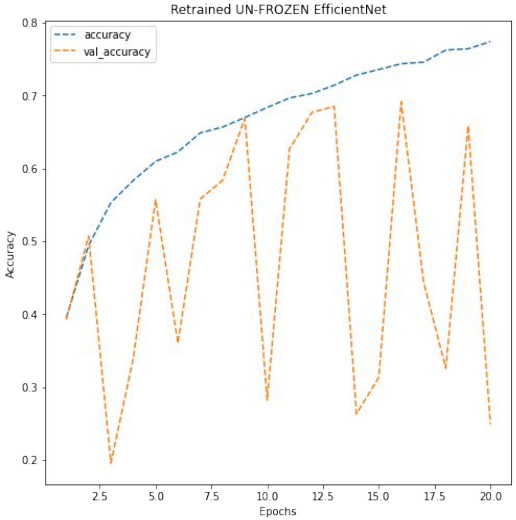
FC = fully connected layer




Final Layer Effnet




FROZEN




UN-FROZEN

# Key Findings - EfficientNet B2

The FROZEN EfficientNet B2 experienced chaotic gyrations in performance called the Vanishing Gradient Problem. There is an array of possible reasons for this issue, initial weights being one of them.

Recent work by Yilmaz and Poli (Neural Networks, 2022) suggest that optimizing initial weights can be an effective antidote. They claim "deep MLPs using sigmoid activation functions can be effectively trained using the standard back-propagation algorithm without experiencing the vanishing gradient problem."

They suggest setting the mean initial weights to max(−1, -8 / number_of_neurons_in_layer).

The UNFROZEN EfficientNet B2 experienced significant issues during training.  Running this model on the training dataset resulted in an exploding gradient problem, often caused by poorly chosen initial weights.

**Resolving these issues will remain a project for a future date.**

Yilmaz, A., & Poli, R. (2022) Successfully and efficiently training deep multi-layer perceptrons with logistic activation function simply requires initializing the weights with an appropriate negative mean. *Neural Networks*, **153**: 87-103. link

# Conclusions

Both the VGG16 and ResNet101 models benefited from the use of the entire body of feature-extraction layers.

VGG16 and ResNet101 model architecture performed even better when pre-trained weights were not used and then trained on the FER2013 dataset. Results approached that of the final CNN model discussed in the original capstone project (link).

Much more work needs to be done on the EfficientNet B2 model to resolve the vanishing and exploding gradient dysfunction seen in this project.

## Future Work & Learning Topics

Optimize the EfficientNet B2 model by applying the use of mean initial weights as per Yilmaz & Poli (link)

Explore the use of Liquid CfC (closed-form Continuous-time) neural network models for “out-of-distribution generalization” that allows the use of pretrained models, as in transfer learning, but without the need for additional training in the new environment/data-field.

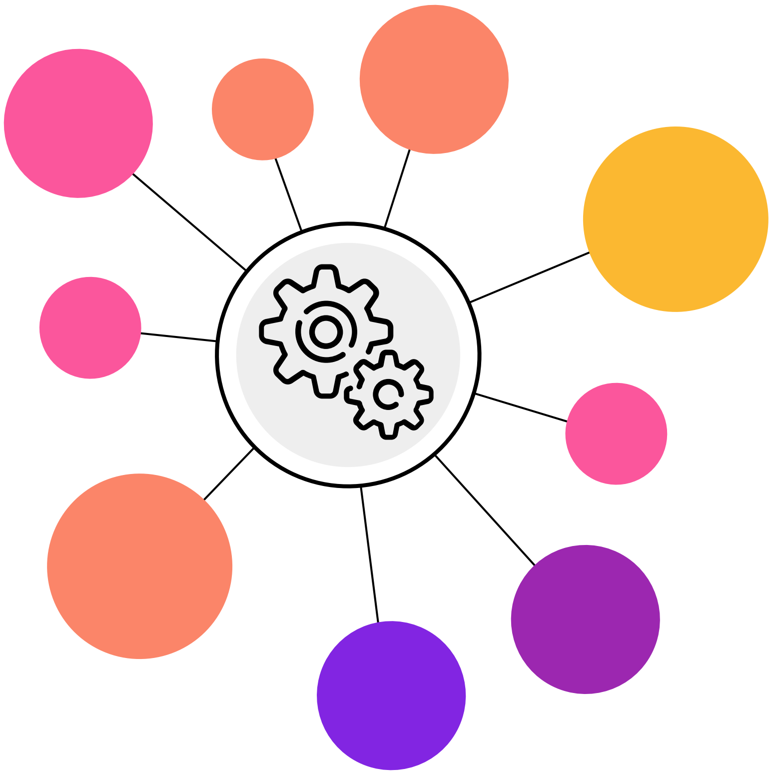
Reference: Hasani, R. et al. (2022) Closed-form continuous-time neural networks. *Nat Mach Intell*

# Appendix

# Original Capstone Presentation

follows

Link to original capstone presentation PDF



# Capstone Project

## Deep Learning Methods for Facial Emotion Recognition

Monica Palacios Boyce, Ph.D.

Key Takeaways

Problem Definition

Proposed Solution Approach

Key Findings & Insights

Recommendations

Next Steps

# Key Takeaways

Objective: Construct a best-fit Convolutional Neural Network (CNN) model that accurately performs multi-class classification for facial emotion recognition.

Must accurately detect four specific emotions in images of people, including: ‘happy’, ‘sad’, ‘neutral’, and ‘surprise’ from the FER 2013 dataset.

Six test CNN models were designed, trained, validated, tuned, optimized, and evaluated.

- Three of these test models included the use of transfer learning.
  - VGG16
  - ResNet V2
  - EfficientNet
- A range of hyperparameters were assessed for positive impact on model performance.

A complex CNN was designed that was able to classify the correct emotion in novel images with approximately **93%** accuracy and generalized well.

# Problem Definition

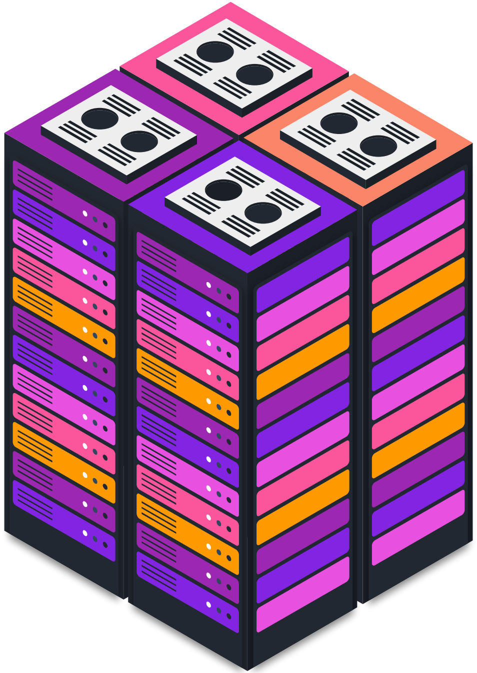
Recognizing accurate emotions in facial images can provide a deeper understanding of the user and situation in which the image was obtained.

Vast amounts of image data is continuously being captured. Many of these images are unlabeled and would require far more people to encode them than are available or feasible.

Convolutional Neural Network models (CNNs) have been developed to process image data to learn higher order patterns (features) that can yield predictions of value on new images.

Challenges include data quality issues, dataset imbalances due to demographic biases, and the need to train on very large datasets to yield sufficiently accurate performance.

# Proposed Solution Approach



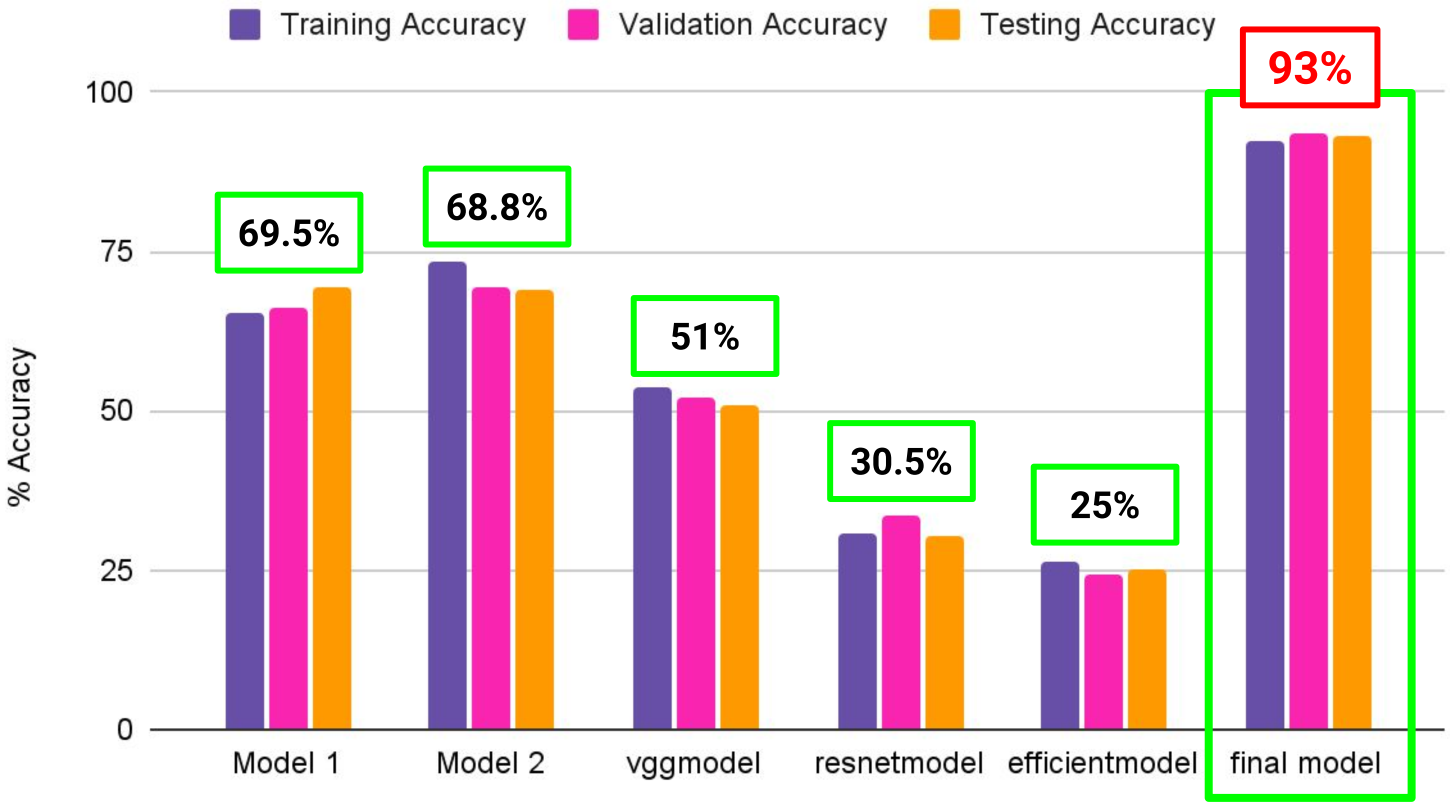
- 5 Convolutional Blocks
  - Convolutional Layer
  - Batch Normalization
  - Leaky ReLU
  - Max Pooling
- Flatten
- 2 Fully Connected Layers
  - Batch Normalization
  - Dropout
- 1 Fully Connected Output Layer

[Link to detailed model architecture diagram in appendix]

## Key Findings & Insights

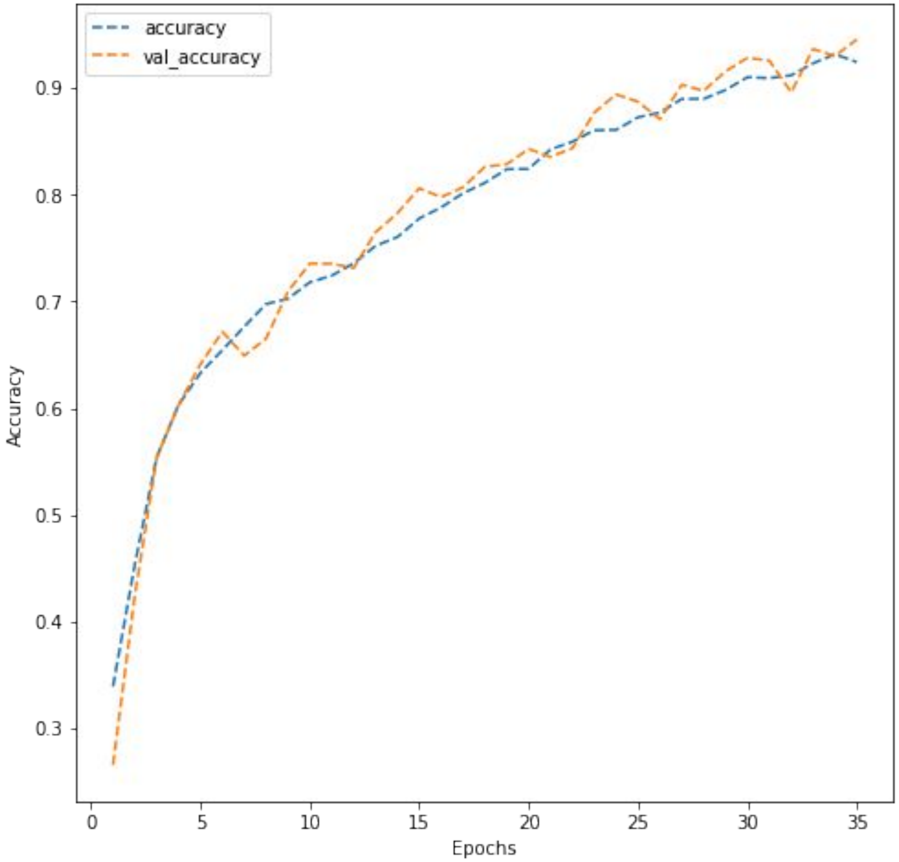
- Tuning focused on data augmentation, optimizers and output layer activation.
- Models that had fewer dropout layers performed better.
  - To effectively build higher order features (object filters), data density needs to remain intact during the feature extraction phase (convolutional layers)
  - Having dropout layers in the classification (fully connected) blocks did not degrade performance.
- Data augmentation strategies did not yield improved performance.
- The final model has a **93%** testing accuracy and good generalized performance.

# Comparison of Performance Across Six Models in this project




Link to data table in appendix

# Final model performance on training and validation data




Illustrates “generalization” of performance over the duration of model training.

Training (blue line) and validation (orange line) accuracy follow very similar paths = generalization is sufficient.

# Recommendations

Further optimize candidate model to improve informative feature extraction by training on larger datasets with higher label fidelity and demographic diversity.

Potential benefits:

- Higher emotion recognition accuracy across a more diverse demographic spectrum
- Any product using this optimized model would be competitive in global markets

Revisit the use of transfer learning to take advantage of the feature extraction layers of pre-trained CNN models, which only continue to grow ever more powerful.

Explore recent advances including, dual-channel CNN architecture that first identifies a region of interest (ROI) and then applies higher resolution feature extraction to the "pre-qualified" ROIs.

# Next Steps

## Mitigate low training performance issues by:

- Increasing the size of the dataset
- Increasing the accuracy of dataset labeling
- Correction of bias by balancing demographic factors (equal representation of genders, ages, and racial phenotypes)

## Further optimization using larger image datasets, such as:

- ImageNet (>14 million annotated images)
- CelebA (>202,000 annotated images)
- FFHQ (Flickr-Faces-HQ, 70,000 high resolution diverse image set)

## Risks

Ethical risks regarding privacy and ownership issues will require an open societal level discourse that should be considered a necessary component of any development plan.

## Challenges

Balancing computational costs required to train development models on very large datasets against potential benefits.

## Opportunities

A sampling of business use-cases for FER:

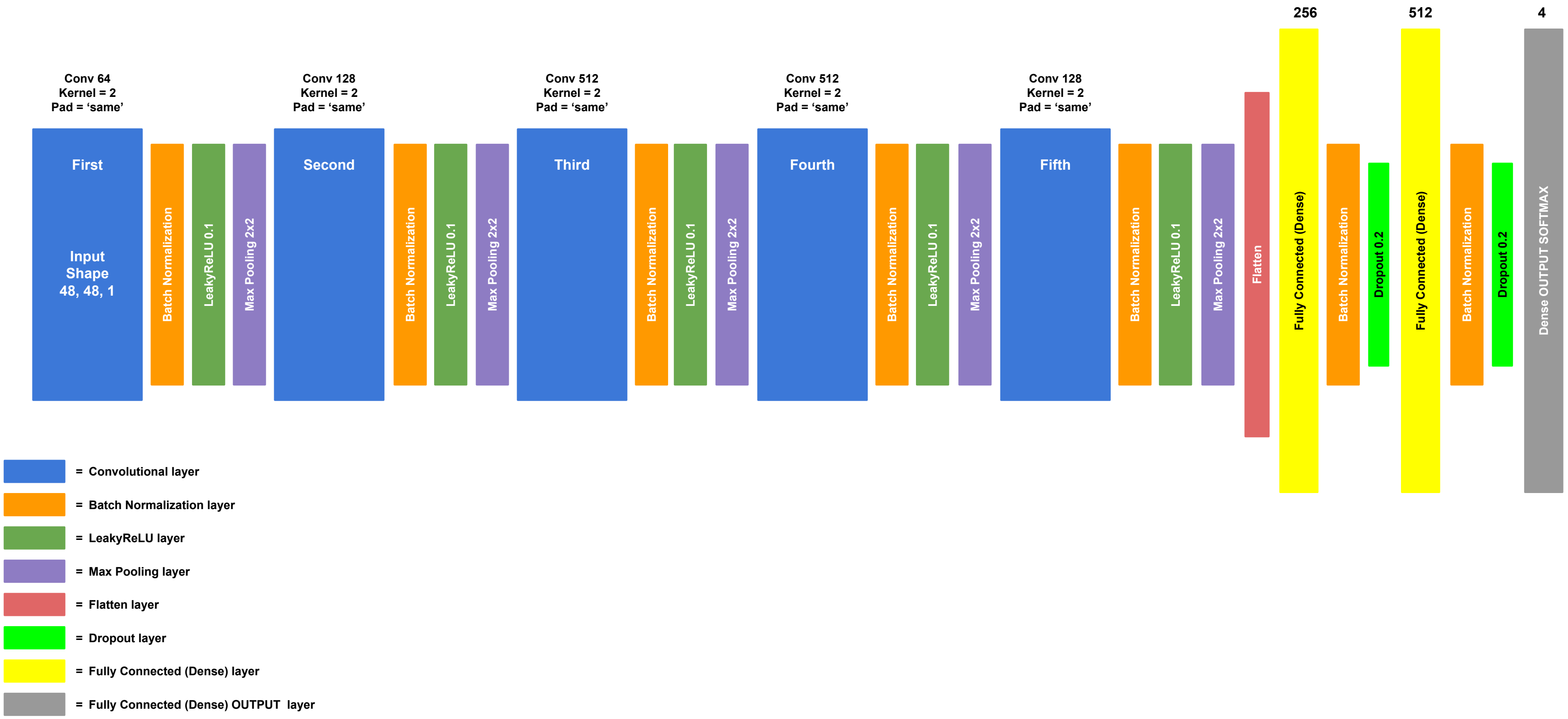
- Capturing metrics of student engagement in online education
- Psychological analysis of job applicants by human resource groups during hiring
- Optimizing personalized learning milieu through the analysis of not only visual facial features but EEG data as a neurological emotion-ground-truth reference.

# Appendix

# Comparison of Performance Across Six Models in this project

| Model | Training Accuracy | Validation Accuracy | Testing Accuracy |
|---|---|---|---|
| Model 1 | 65.4% | 66.1% | 69.5% |
| Model 2 | 73.5% | 69.4% | 68.8% |
| vggmodel | 53.8% | 52.1% | 51% |
| resnetmodel | 30.6% | 33.4% | 30.5% |
| efficientmodel | 26.3% | 24.4% | 25% |
| final model (rgb) | 92.4% | 93.6% | 93.3% |

# Final Model Design



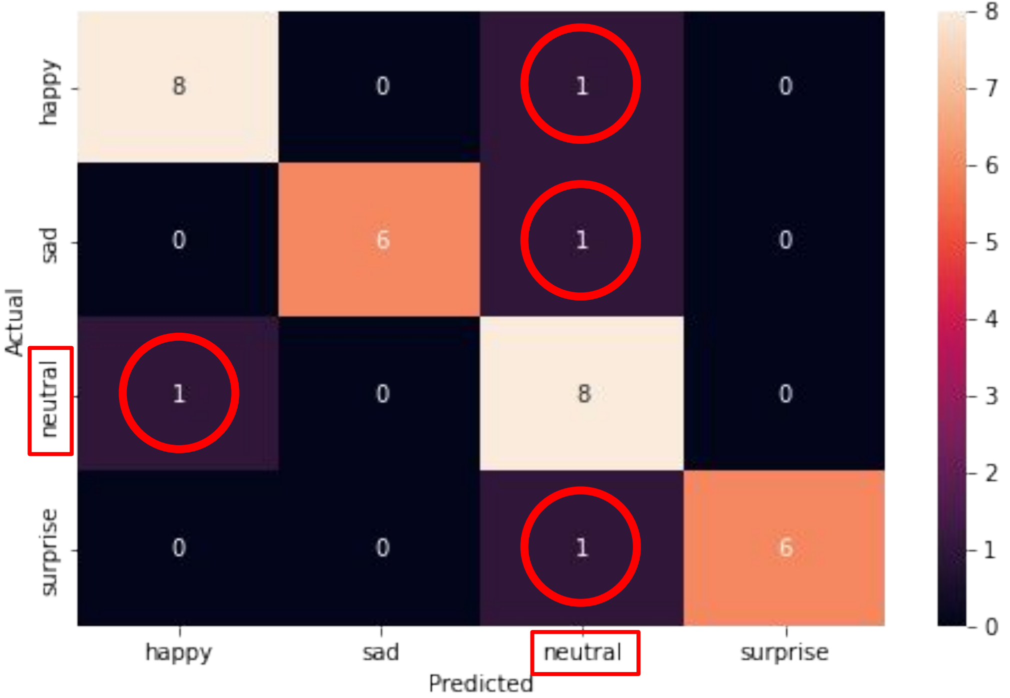

Link back to main slide

# Which emotions are misclassified by the model?

Incidence of prediction errors by the model on 128 test images, 32 in each class.

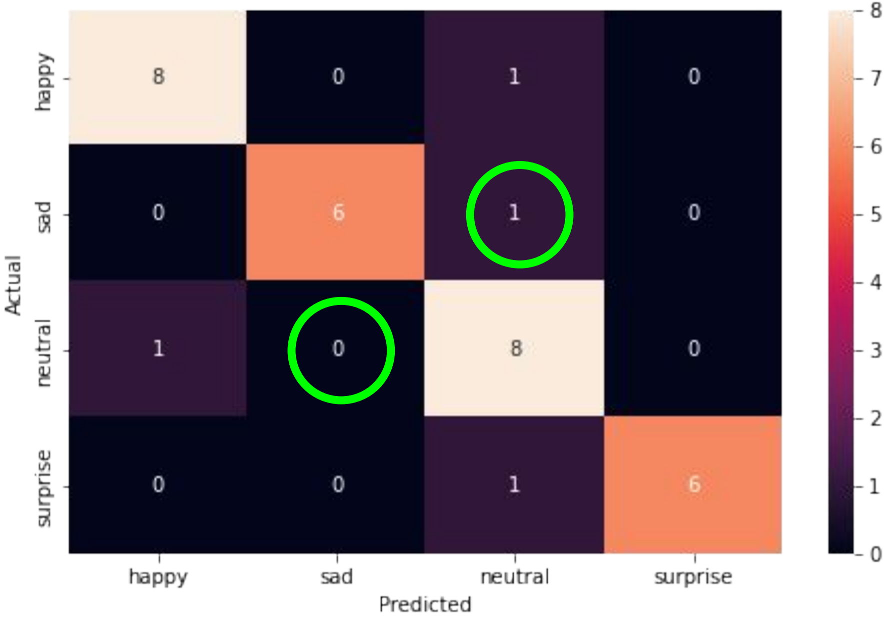
Observation: Those boxes showing an error are associated with the 'neutral' emotion.

Insight: This illustrates the difficulty of classifying an "emotion" from a neutral face.



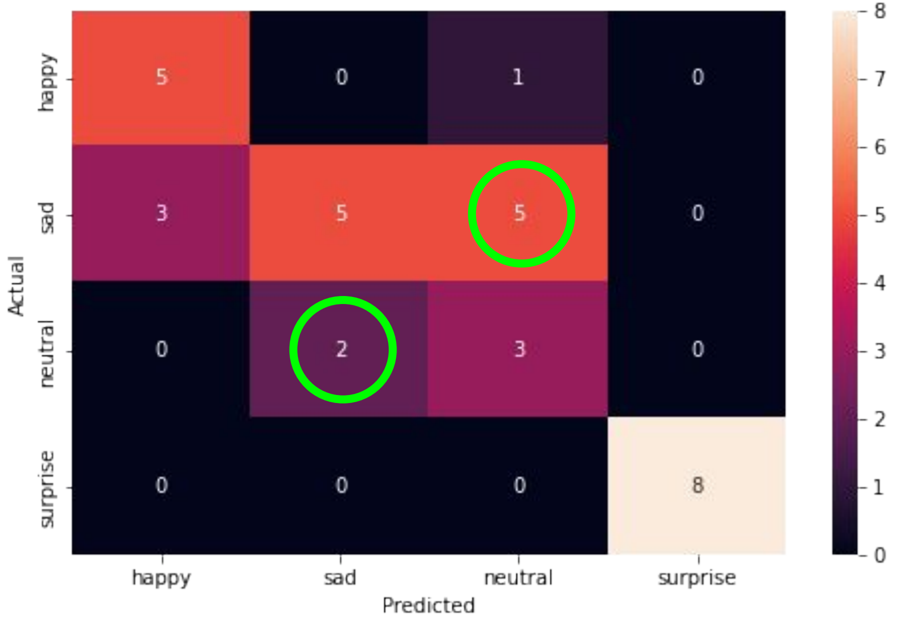
# Comparing ‘rgb’ to ‘grayscale’ final model design

Final Model - RGB

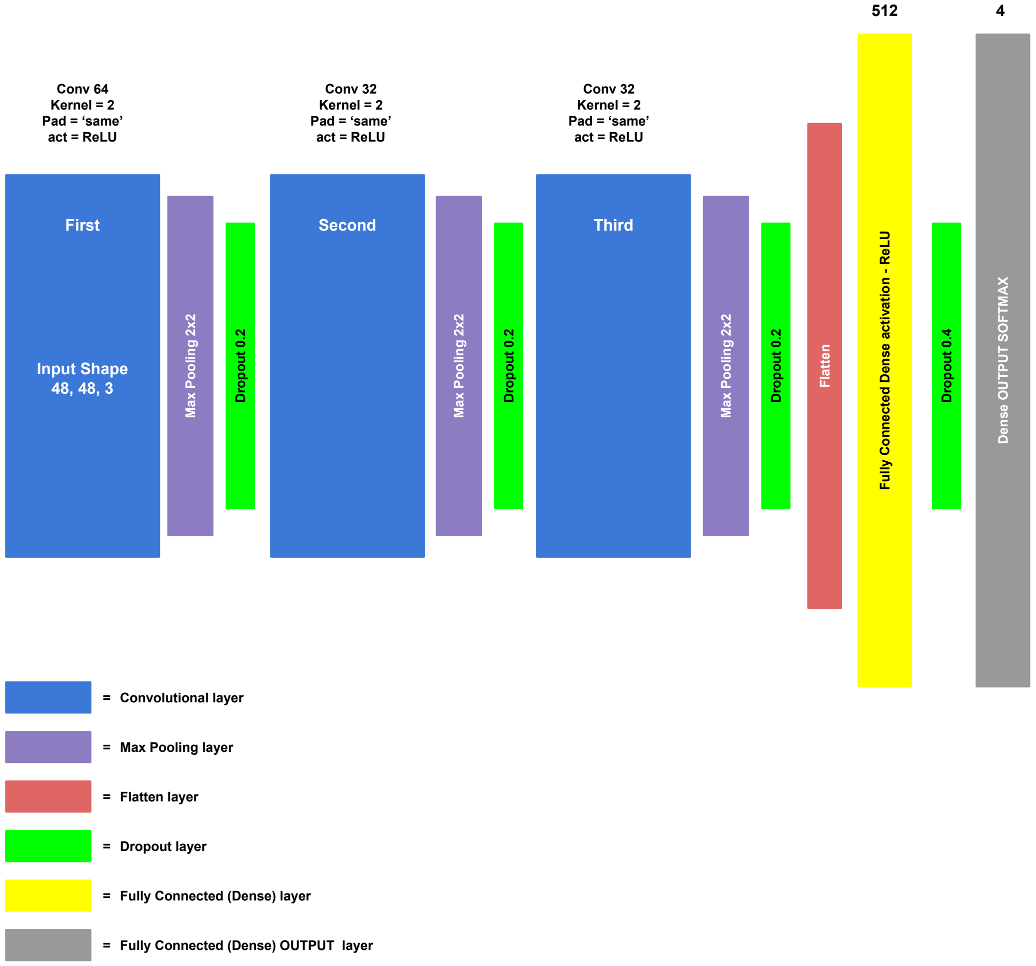



Improved Performance

Final Model - grayscale

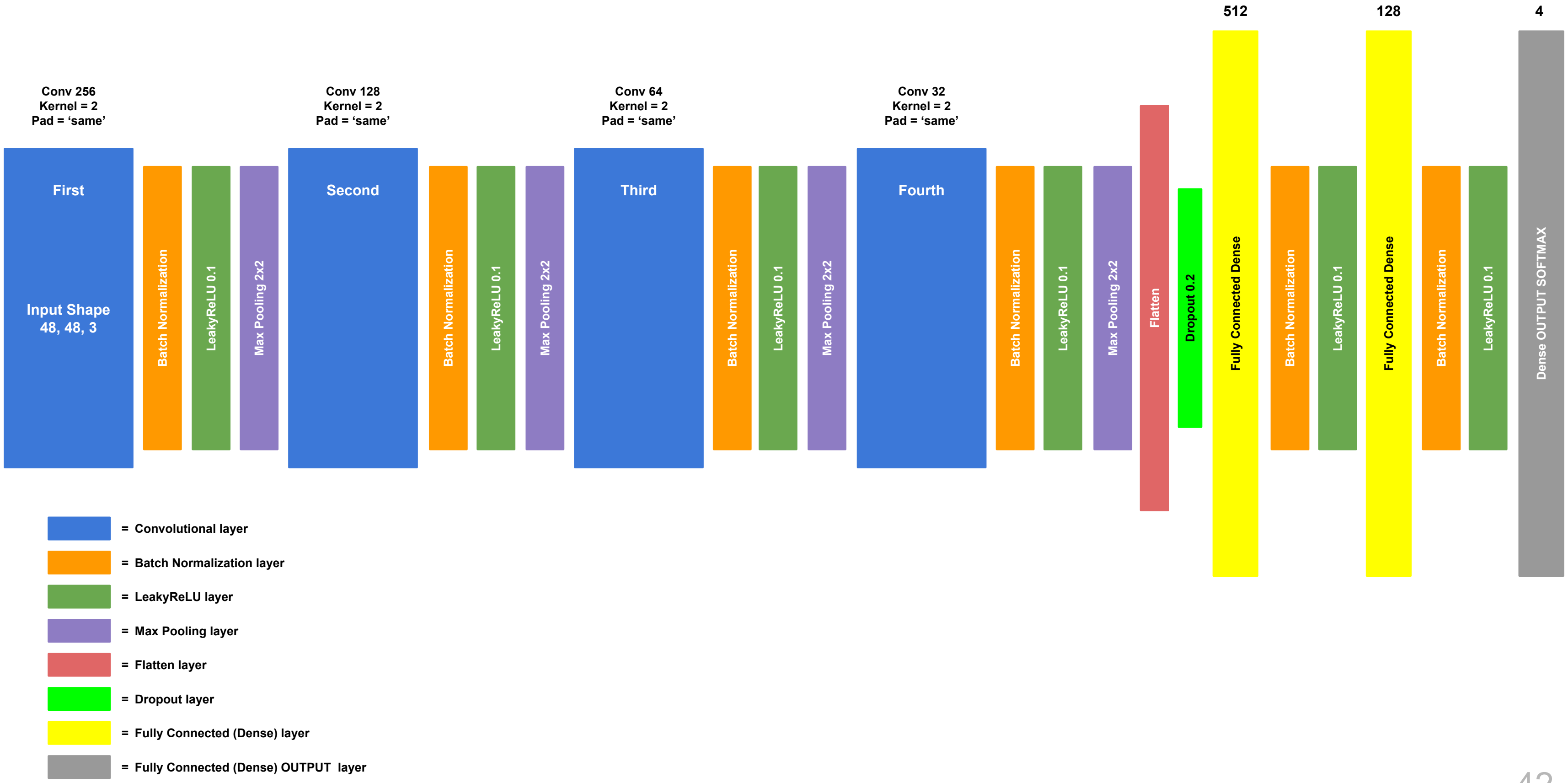


# Model 1



Conv 64
Kernel = 2
Pad = ‘same’
act = ReLU
Conv 32
Kernel = 2
Pad = ‘same’
act = ReLU
Conv 32
Kernel = 2
Pad = ‘same’
act = ReLU
512
4
First
Input Shape
48, 48, 3
Max Pooling 2x2
Dropout 0.2
Second
Max Pooling 2x2
Dropout 0.2
Third
Max Pooling 2x2
Dropout 0.2
Flatten
Fully Connected Dense activation - ReLU
Dropout 0.4
Dense OUTPUT SOFTMAX
= Convolutional layer
= Max Pooling layer
= Flatten layer
= Dropout layer
= Fully Connected (Dense) layer
= Fully Connected (Dense) OUTPUT layer

# Model 2

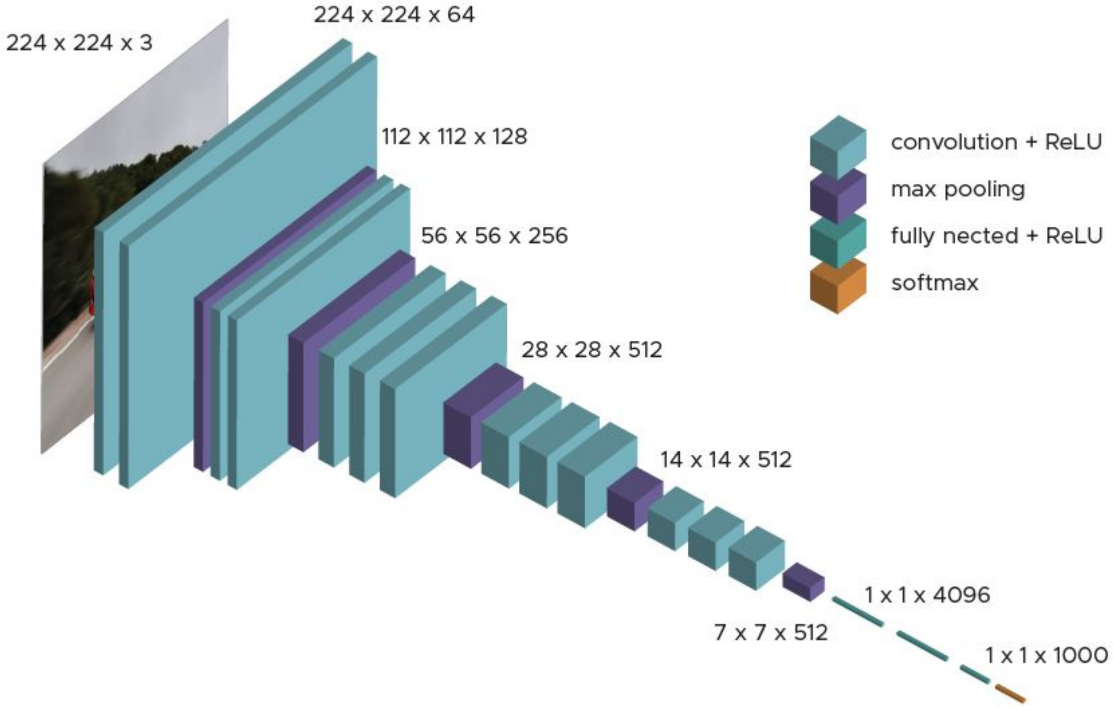


Conv 256
Kernel = 2
Pad = 'same'
First
Input Shape
48, 48, 3
Batch Normalization
LeakyReLU 0.1
Max Pooling 2x2
Conv 128
Kernel = 2
Pad = 'same'
Second
Batch Normalization
LeakyReLU 0.1
Max Pooling 2x2
Conv 64
Kernel = 2
Pad = 'same'
Third
Batch Normalization
LeakyReLU 0.1
Max Pooling 2x2
Conv 32
Kernel = 2
Pad = 'same'
Fourth
Batch Normalization
LeakyReLU 0.1
Max Pooling 2x2
Flatten
Dropout 0.2
512
Fully Connected Dense
Batch Normalization
LeakyReLU 0.1
128
Fully Connected Dense
Batch Normalization
LeakyReLU 0.1
4
Dense OUTPUT SOFTMAX
= Convolutional layer
= Batch Normalization layer
= LeakyReLU layer
= Max Pooling layer
= Flatten layer
= Dropout layer
= Fully Connected (Dense) layer
= Fully Connected (Dense) OUTPUT layer

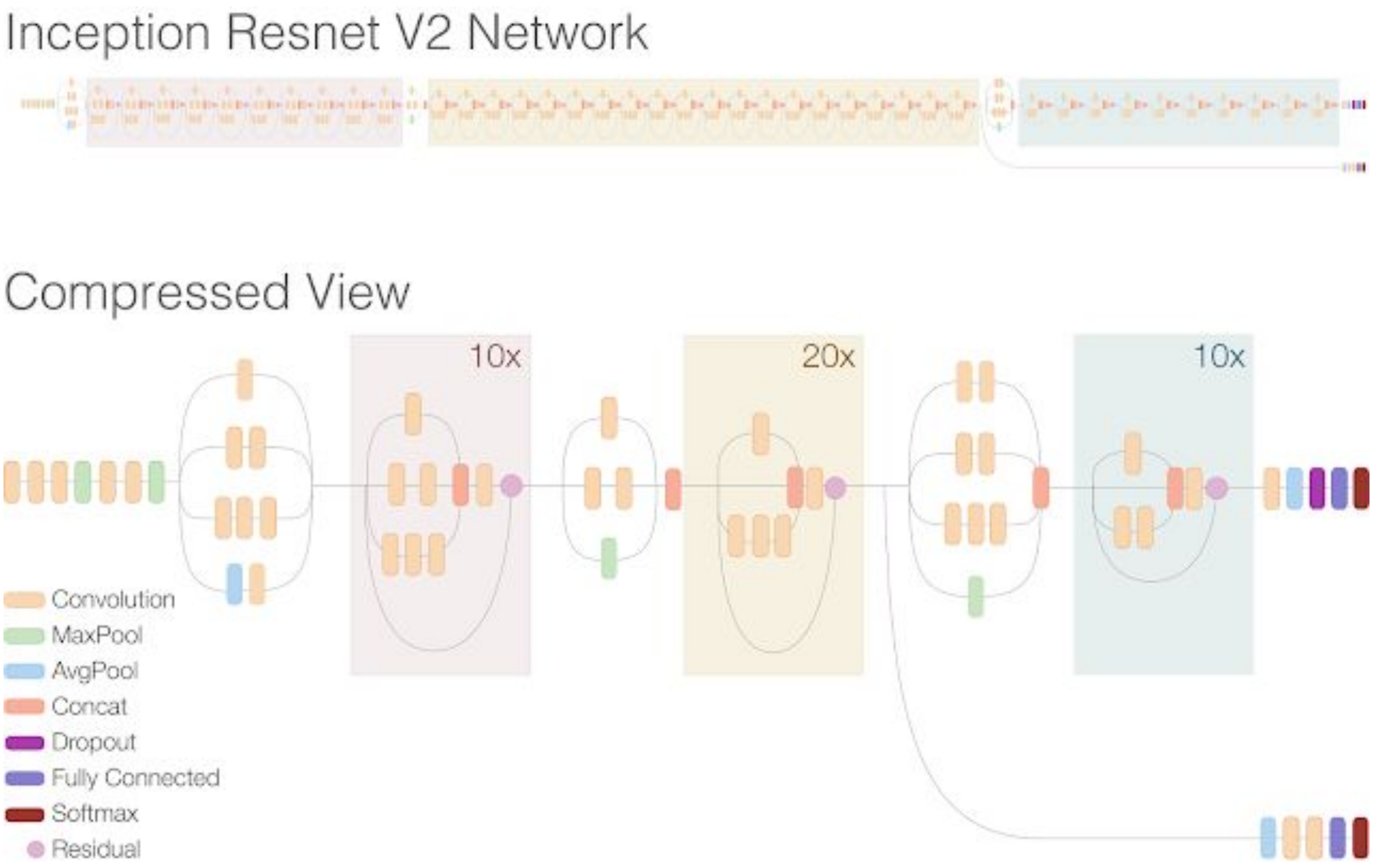
# VGG16

**16 layers of VGG16**

1.Convolution using 64 filters
2.Convolution using 64 filters + Max pooling
3.Convolution using 128 filters
4. Convolution using 128 filters + Max pooling
5. Convolution using 256 filters
6. Convolution using 256 filters
7. Convolution using 256 filters + Max pooling
8. Convolution using 512 filters
9. Convolution using 512 filters
10. Convolution using 512 filters+Max pooling
11. Convolution using 512 filters
12. Convolution using 512 filters
13. Convolution using 512 filters+Max pooling
14. Fully connected with 4096 nodes
15. Fully connected with 4096 nodes
16. Output layer with Softmax activation with 1000 nodes.



# ResNet V2


Inception Resnet V2 Network
Compressed View
10x
20x
10x
Convolution
MaxPool
AvgPool
Concat
Dropout
Fully Connected
Softmax
Residual

# EfficientNet