

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



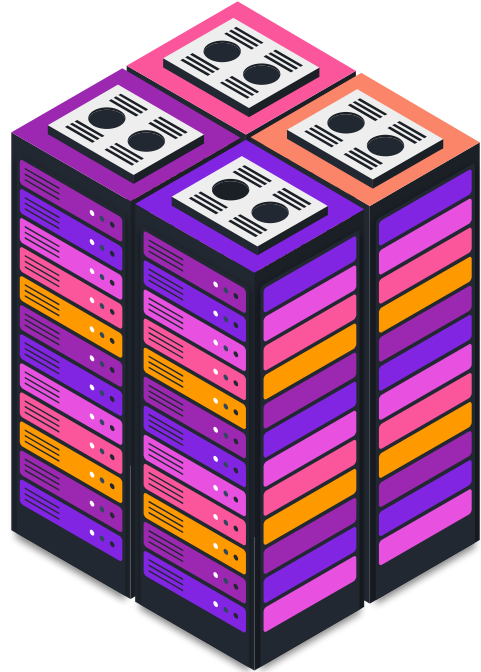
Recognizing accurate emotions in facial images can provide a deeper understanding of the user and situation in which the image was obtained.

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

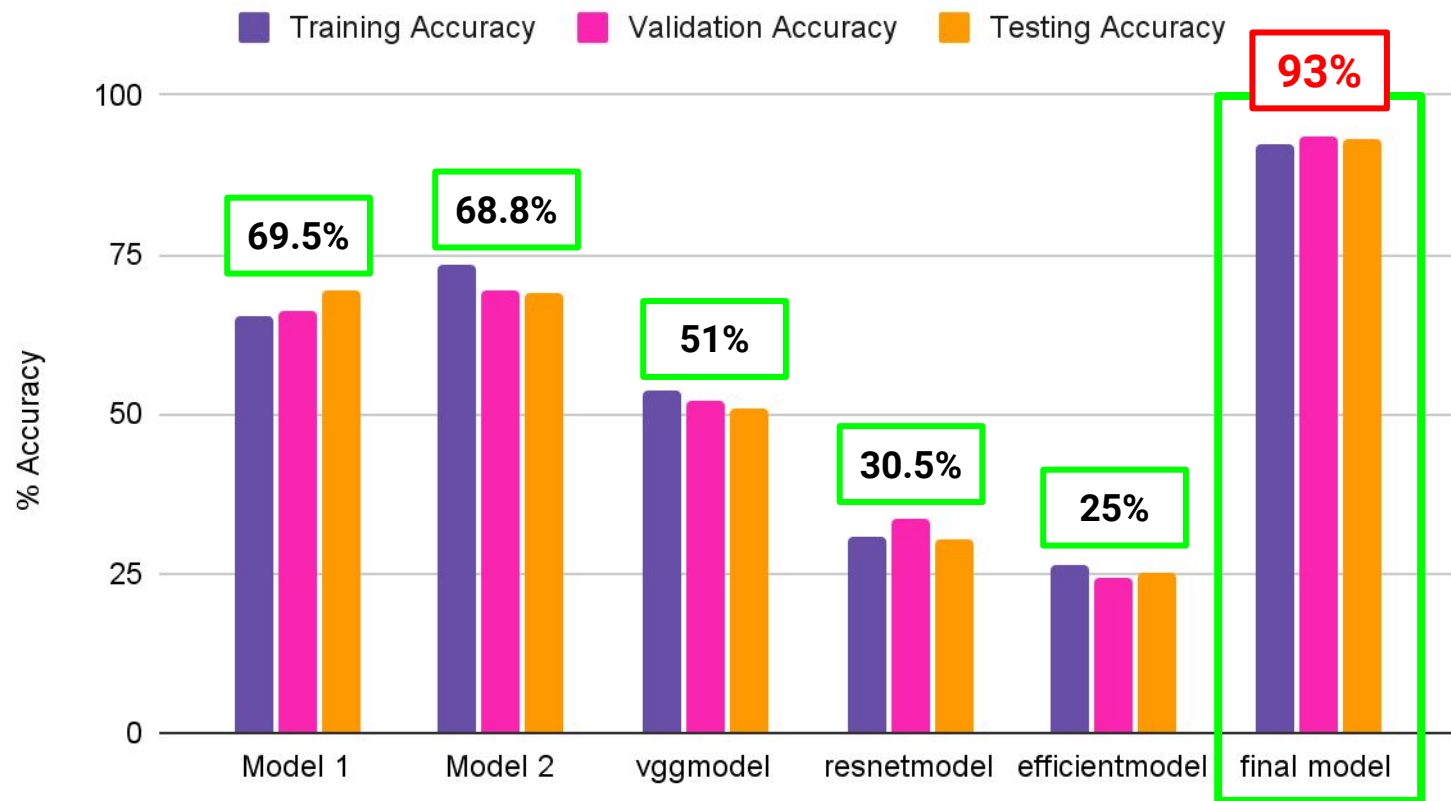
1 Fully Connected Output Layer

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.

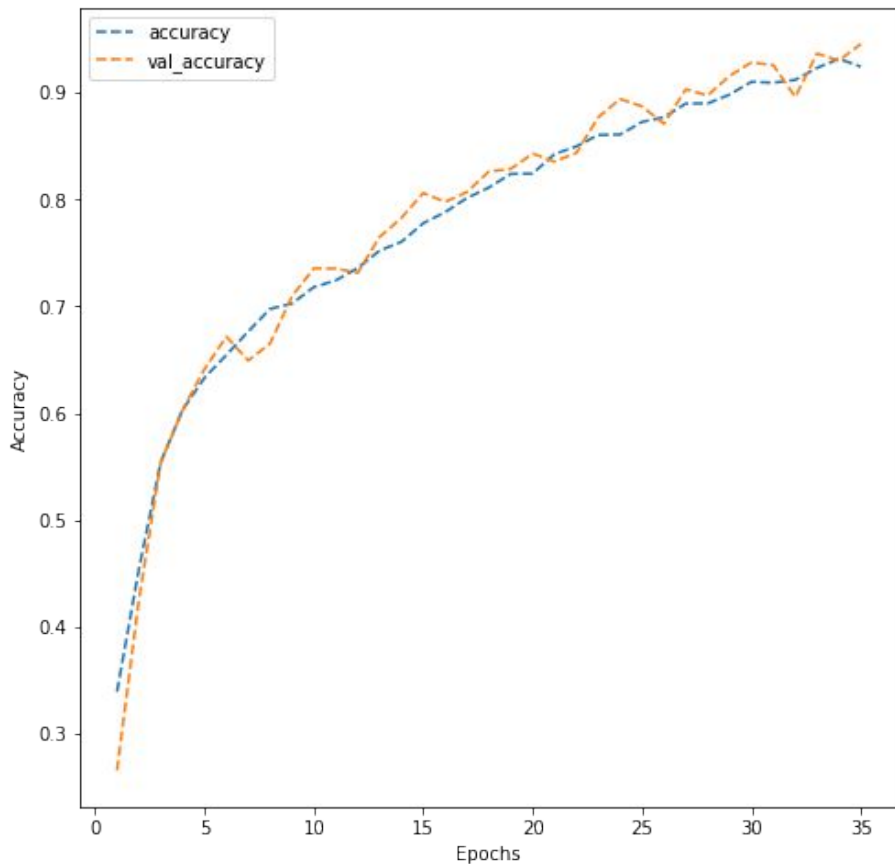
Comparison of Performance Across Six Models in this project



[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.

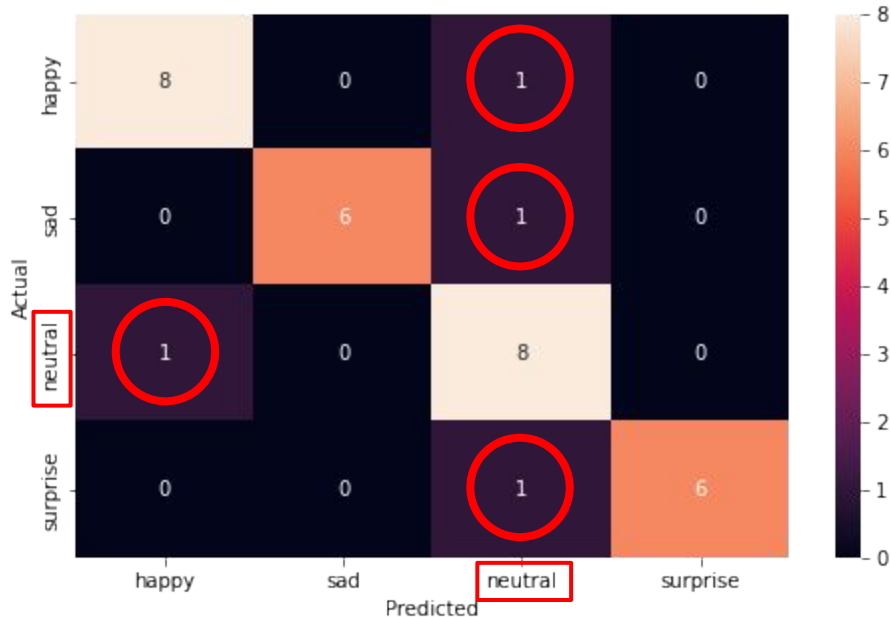


Which emotions are misclassified by the model?

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.



Recommendations



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

Potential benefits:

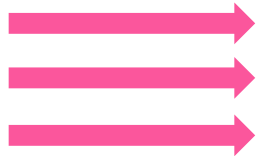
- Higher emotion recognition accuracy across a more diverse demographic spectrum
- Any product using this optimized model would be competitive in global markets

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.



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.



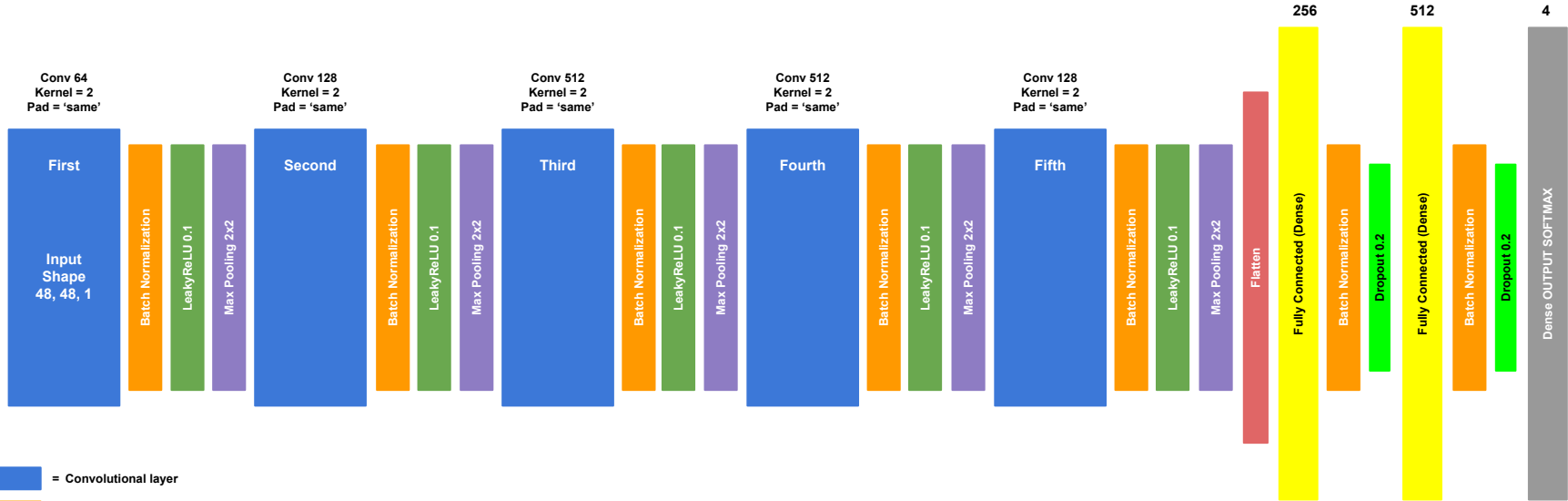
Appendix



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

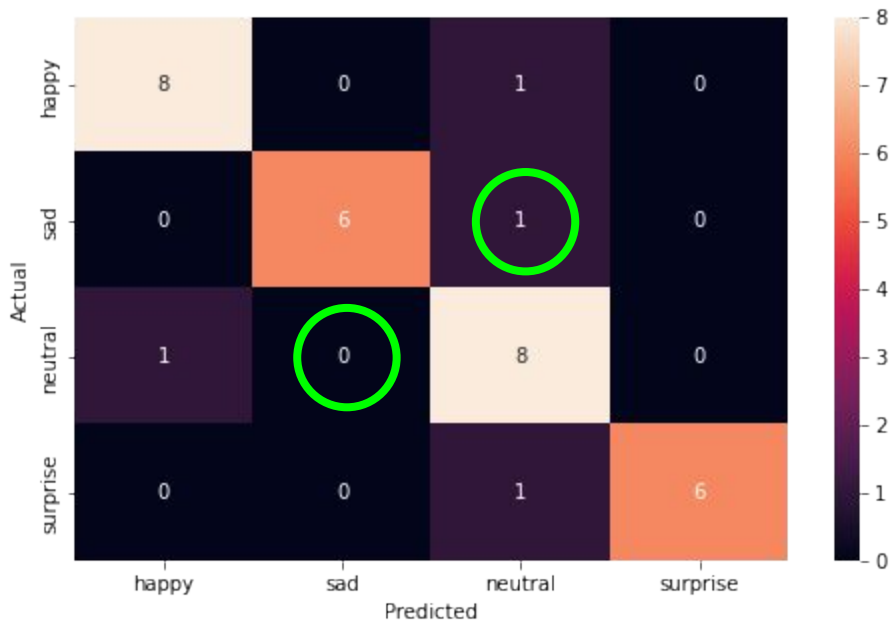


- = Convolutional layer
- = Batch Normalization layer
- = LeakyReLU layer
- = Max Pooling layer
- = Flatten layer
- = Dropout layer
- = Fully Connected (Dense) layer
- = Fully Connected (Dense) OUTPUT layer

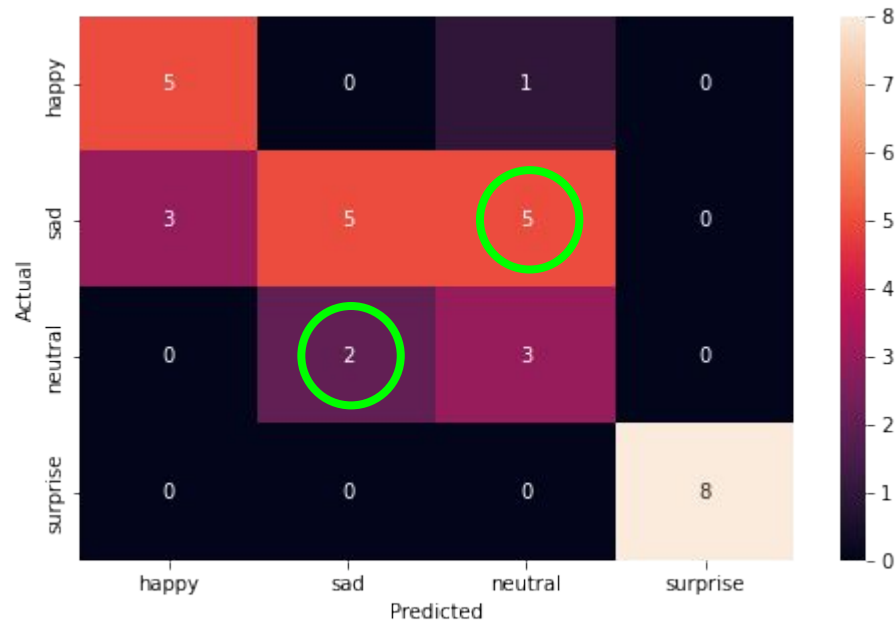


Comparing 'rgb' to 'grayscale' final model design

Final Model - RGB

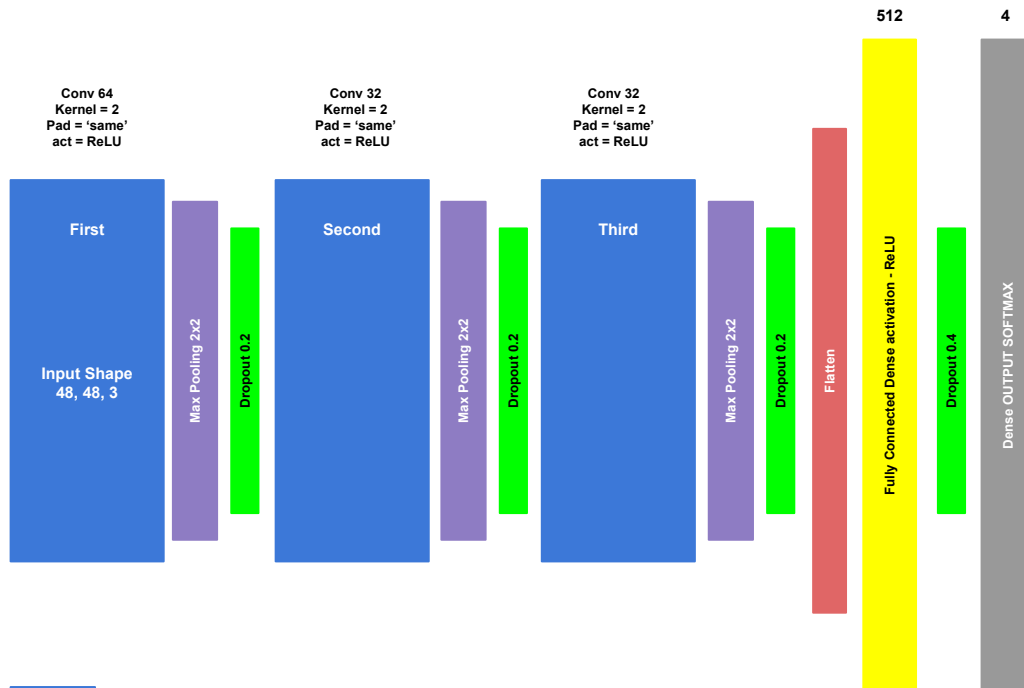








Final Model - grayscale



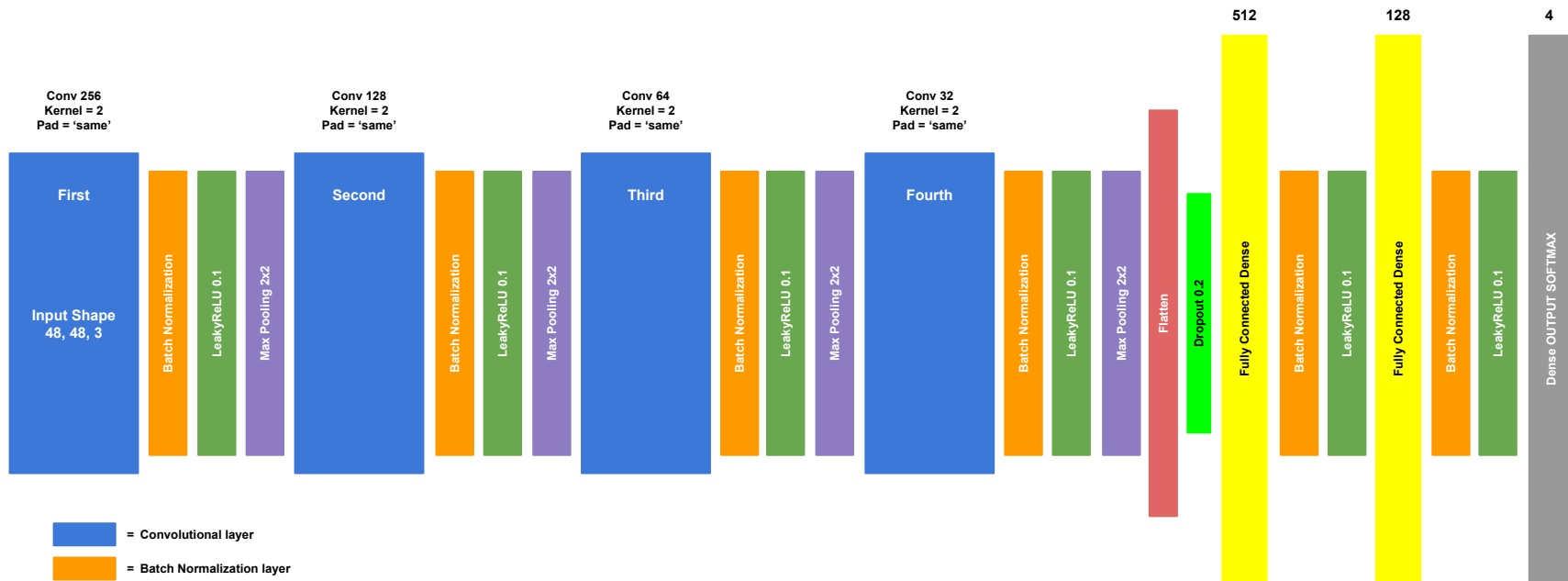
Improved Performance

Model 1



-  = Convolutional layer
-  = Max Pooling layer
-  = Flatten layer
-  = Dropout layer
-  = Fully Connected (Dense) layer
-  = Fully Connected (Dense) OUTPUT layer

Model 2

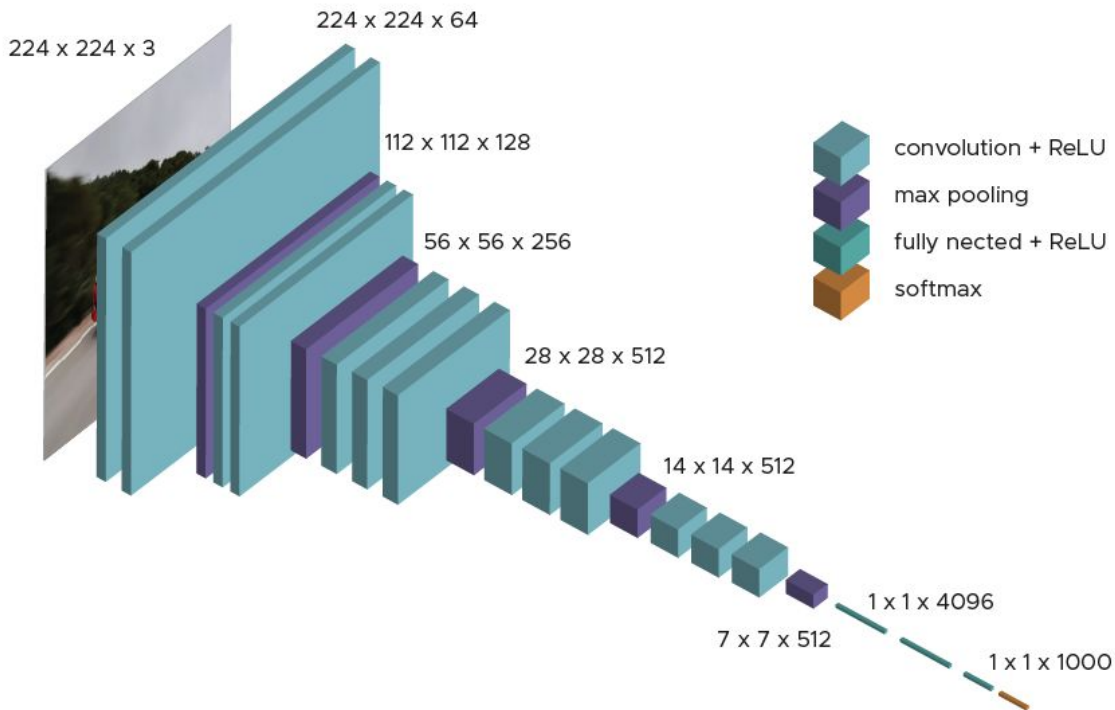


- = Convolutional layer
- = Batch Normalization layer
- = LeakyReLU layer
- = Max Pooling layer
- = Flatten layer
- = Dropout layer
- = Fully Connected (Dense) layer
- = Fully Connected (Dense) OUTPUT layer

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.

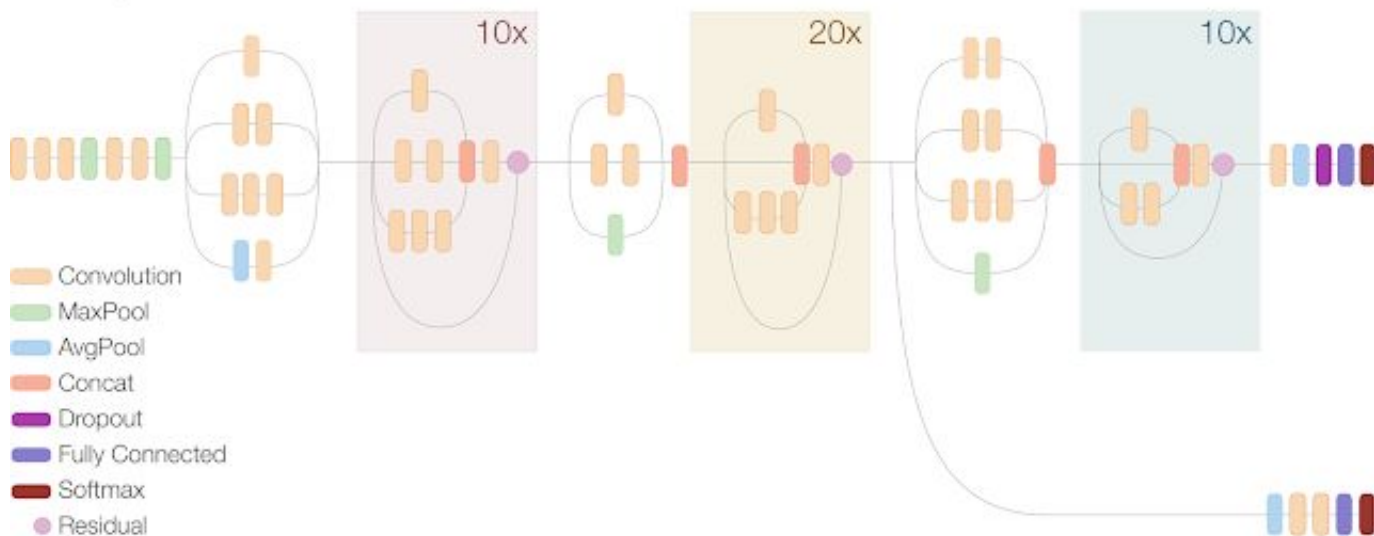


ResNet V2

Inception Resnet V2 Network



Compressed View



EfficientNet

