ML Final Project

110511010 楊育陞 110511067 葉哲伍

Content

01

Train of Thought

Dataset - Observation

- Very small dataset
- Balance class distribution

What methods should we use?

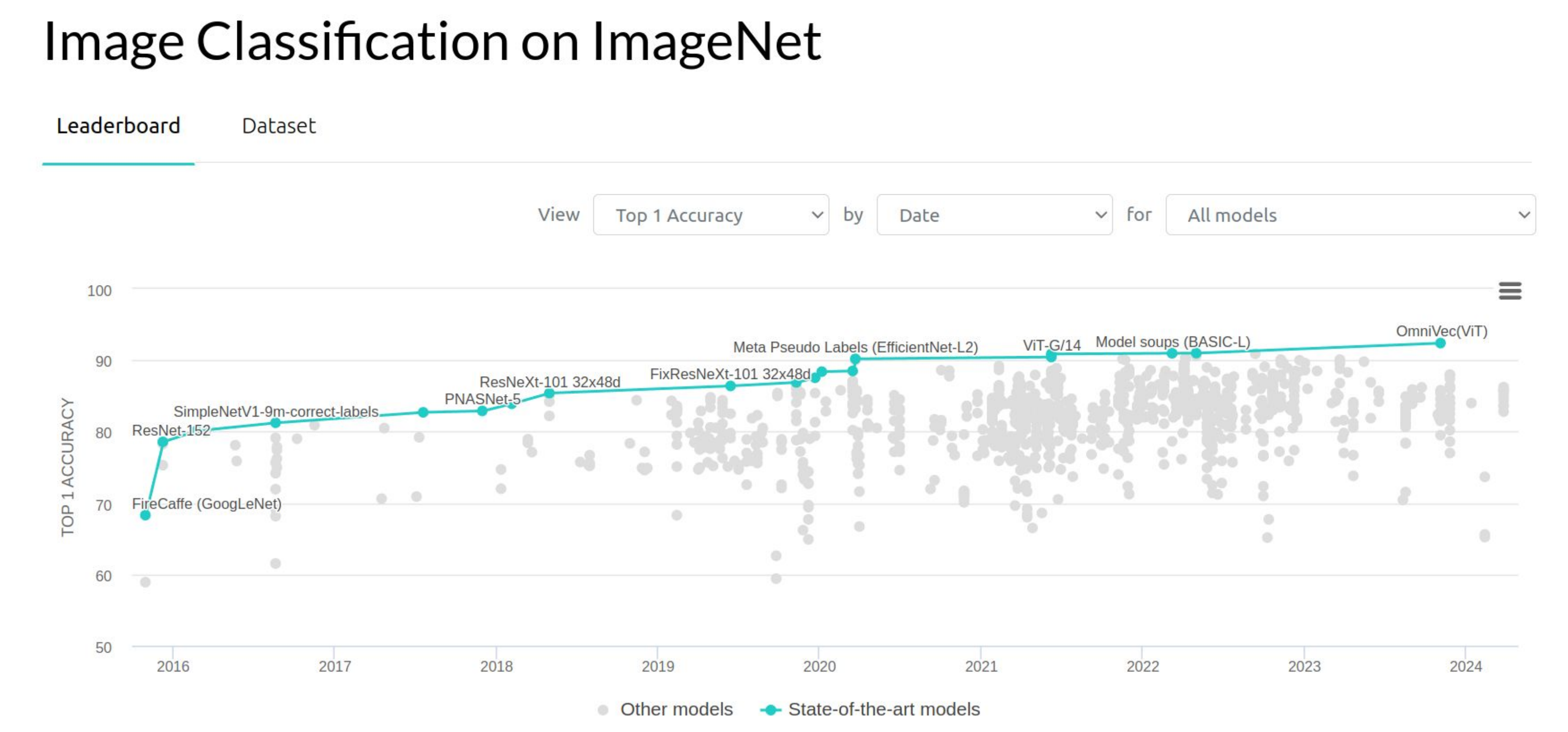
- Binary Logistic Regression
- Support Vector Machine (SVM)
- K-Nearest Neighbor (KNN)
- Neural Network

The Raw Result

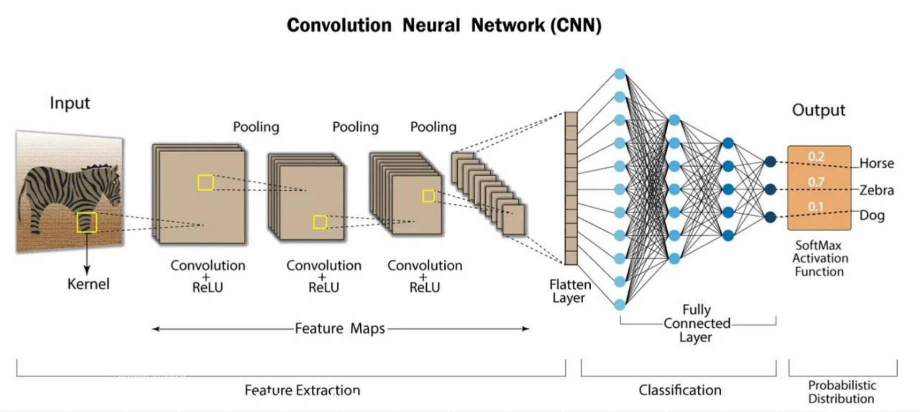
	Logistic Regression	SVM	KNN
Testing Accuracy	52.5%	58.3%	60%

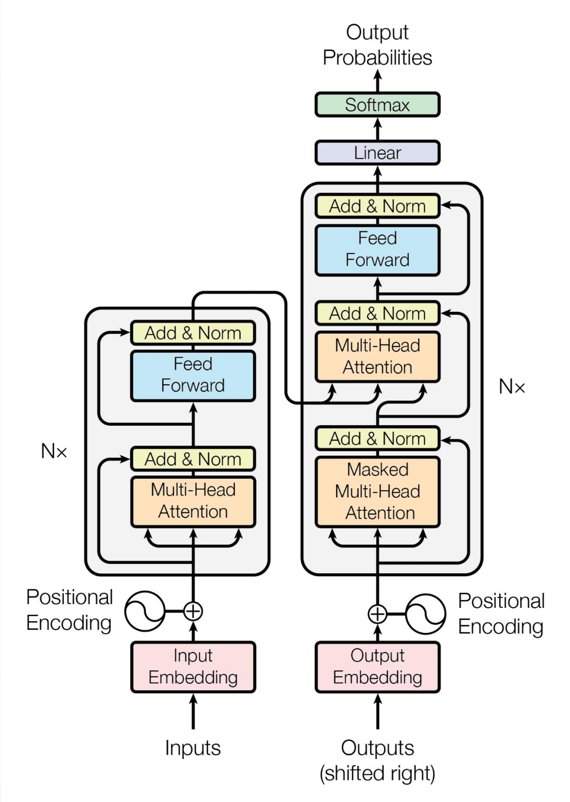
- There are lacking in feature extraction.
- But manual feature extraction is complex and time-consuming.

Survey on image classification models



Why we choose CNN?



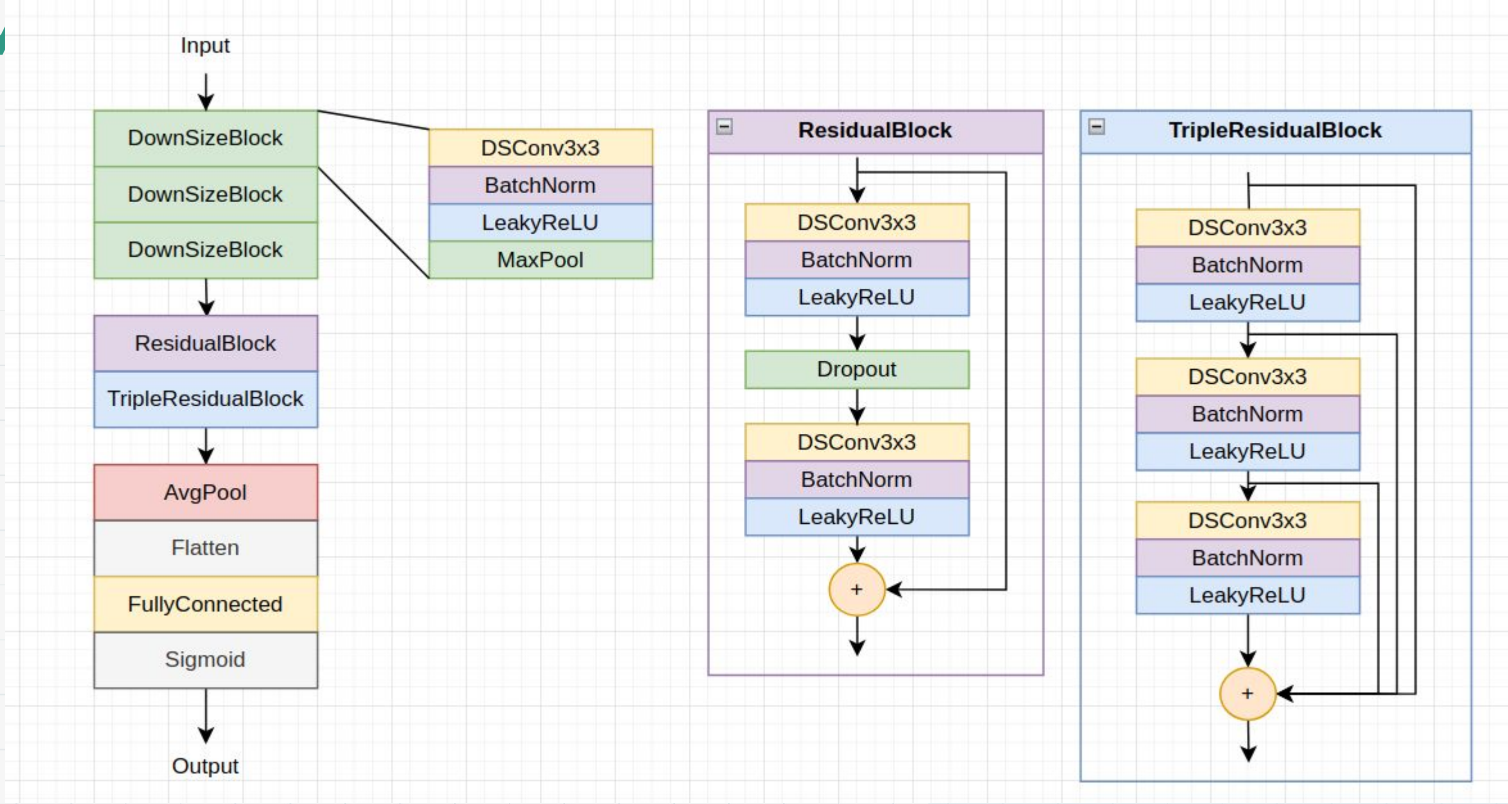


- Considering the size of dataset, and the scale of model.

CNN Construction

- Build from Pre trained model.
    - Image Classification model.
    - Face Detection model.
- Build from scratch.

Conclusion - Custom Lightweight M...



02

Method

Dataset - Observation

- Small dataset
- Balance class distribution
- **People with different orientation, size, pose in picture**
- **Different background and objects**
- **Different light and color distribution**

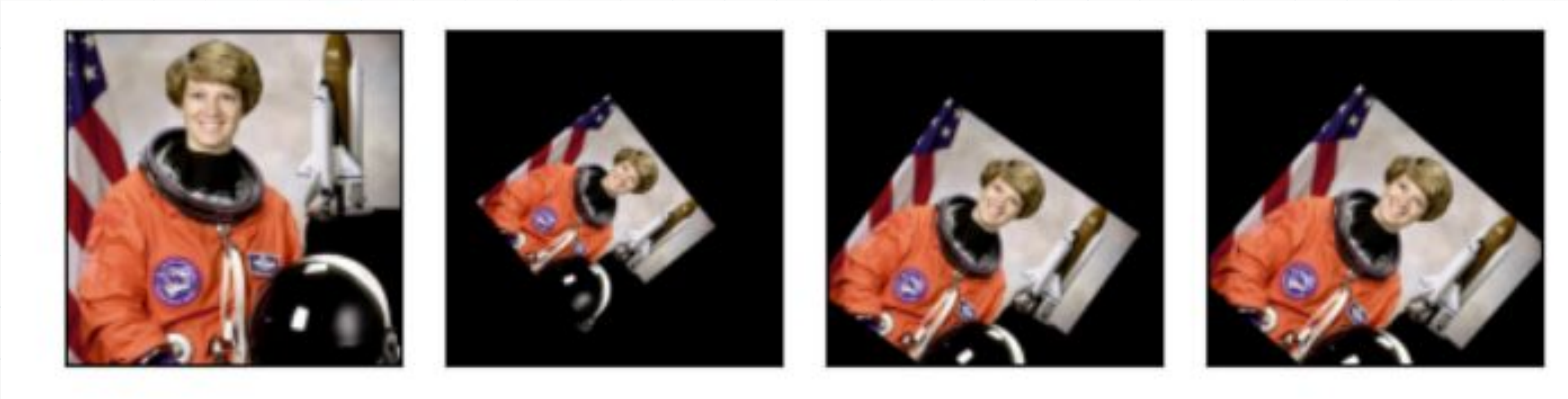



Dataset - Data Augmentation

- People with different orientation, size, pose in picture
    - Random Horizontal Flip

Dataset - Data Augmentation

- **People with different orientation, size, pose in picture**
  - Random Horizontal Flip
  - **Random Affine**



Dataset - Data Augmentation

- **People with different orientation, size, pose in picture**
  - Random Horizontal Flip
  - Random Affine
  - **Random Perspective**

Dataset - Data Augmentation

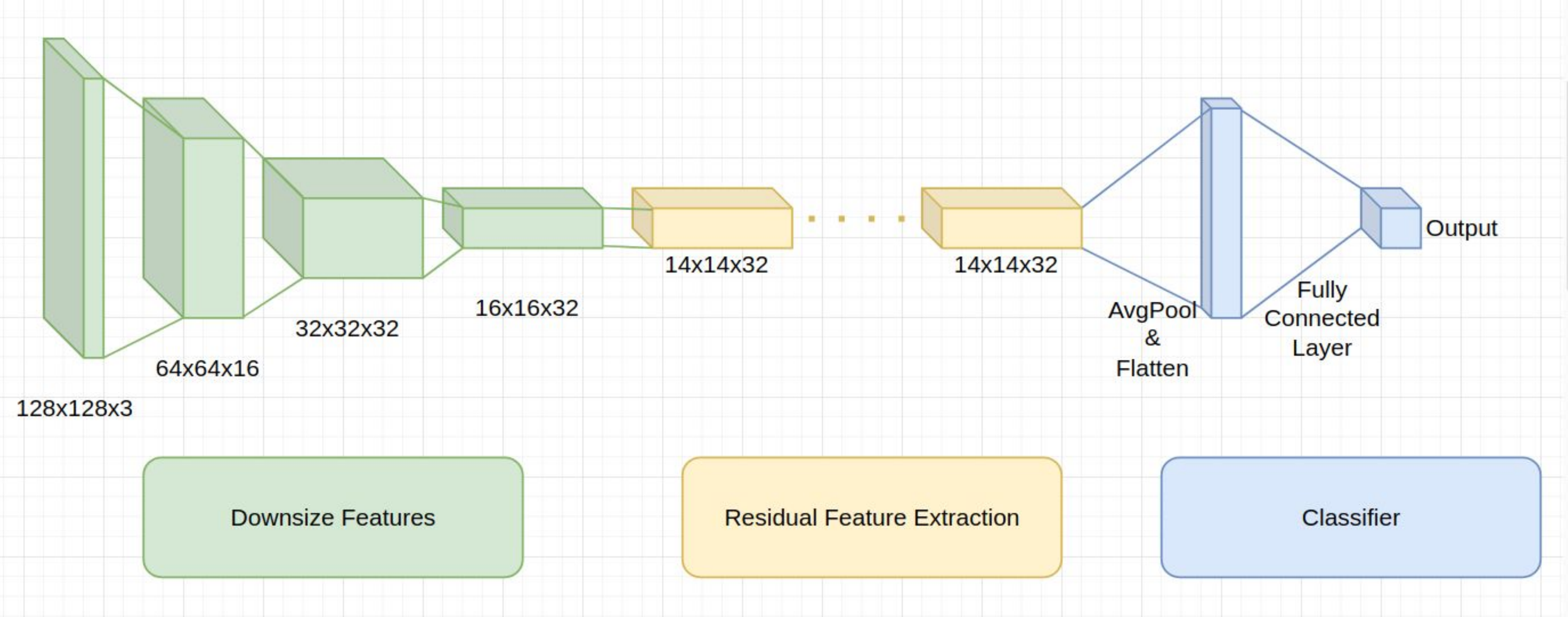
- Different light and color distribution
    - ColorJitter

Dataset - Data Augmentation

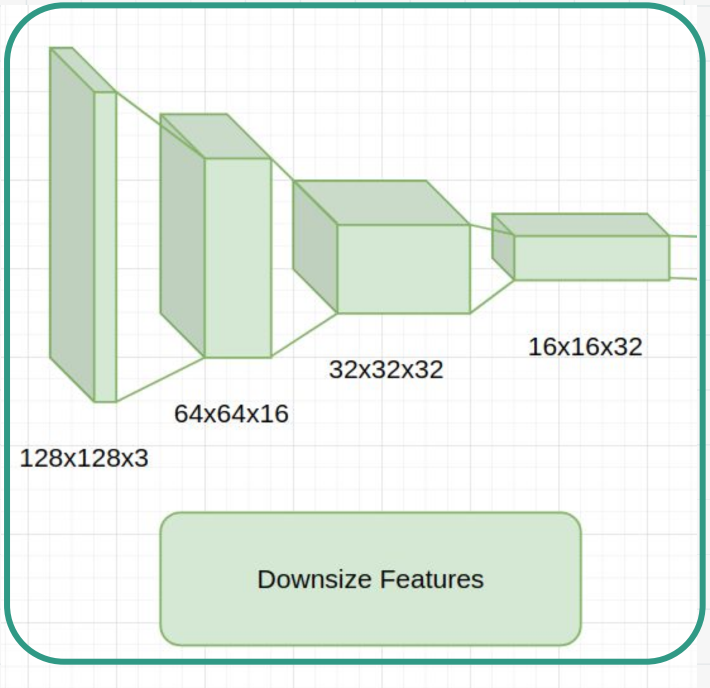
- **Different light and color distribution**
  - ColorJitter
  - **RandomGrayScale**

Model - Features

- Split into 3 parts

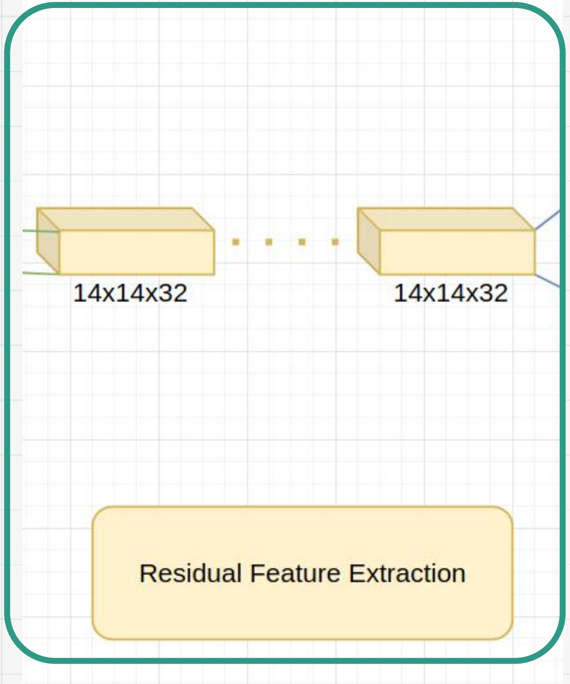


Model - Downsize Features



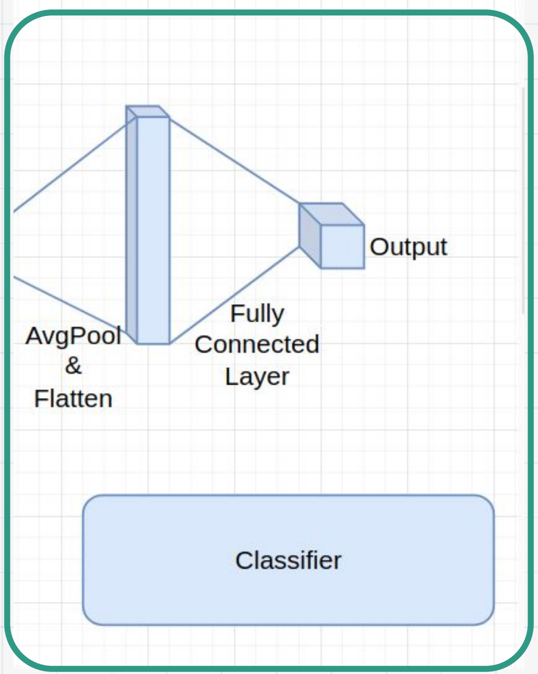
- Downsize features using **pooling layers**
- Reduce computation cost
- Modify the downsize times and the channels to strike the balance between computation cost and accuracy

Model - Residual Feature Extraction



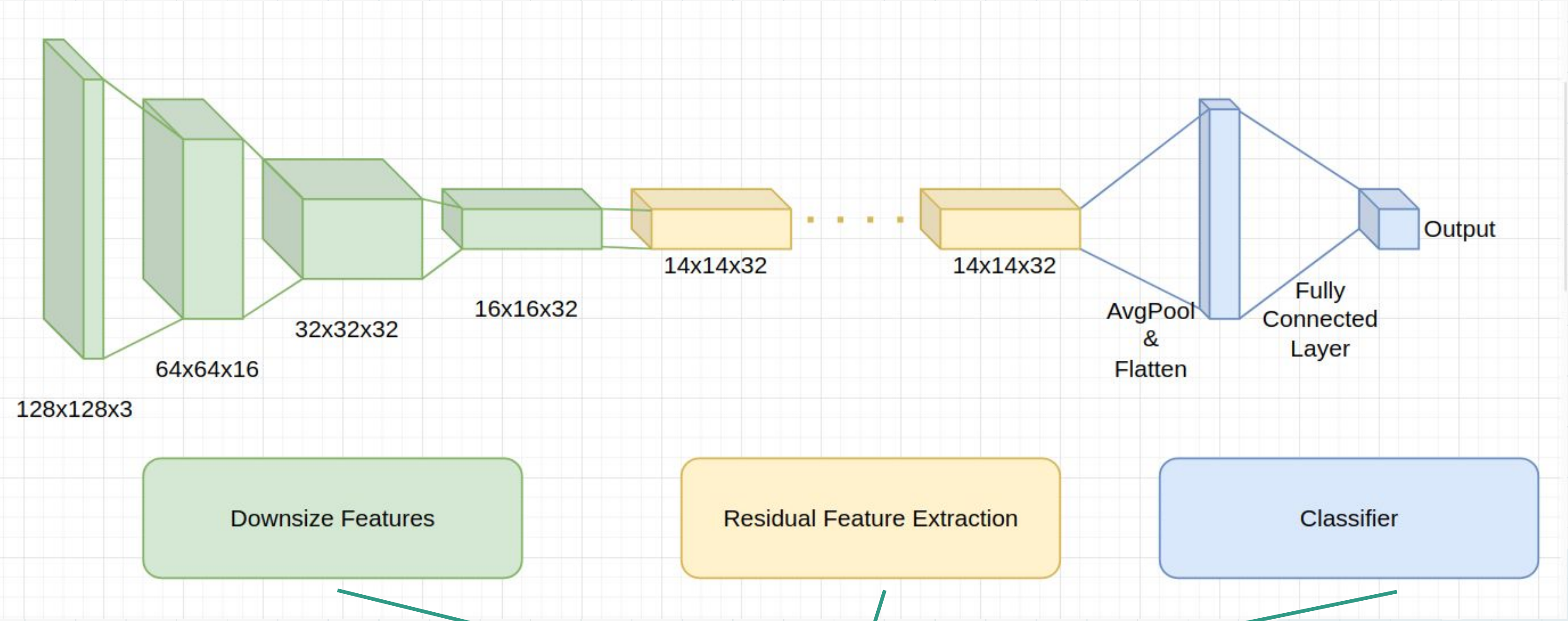
- Feature extraction using **residual layers**
- Modify the number of blocks to adjust the complexity of model

Model - Classifier



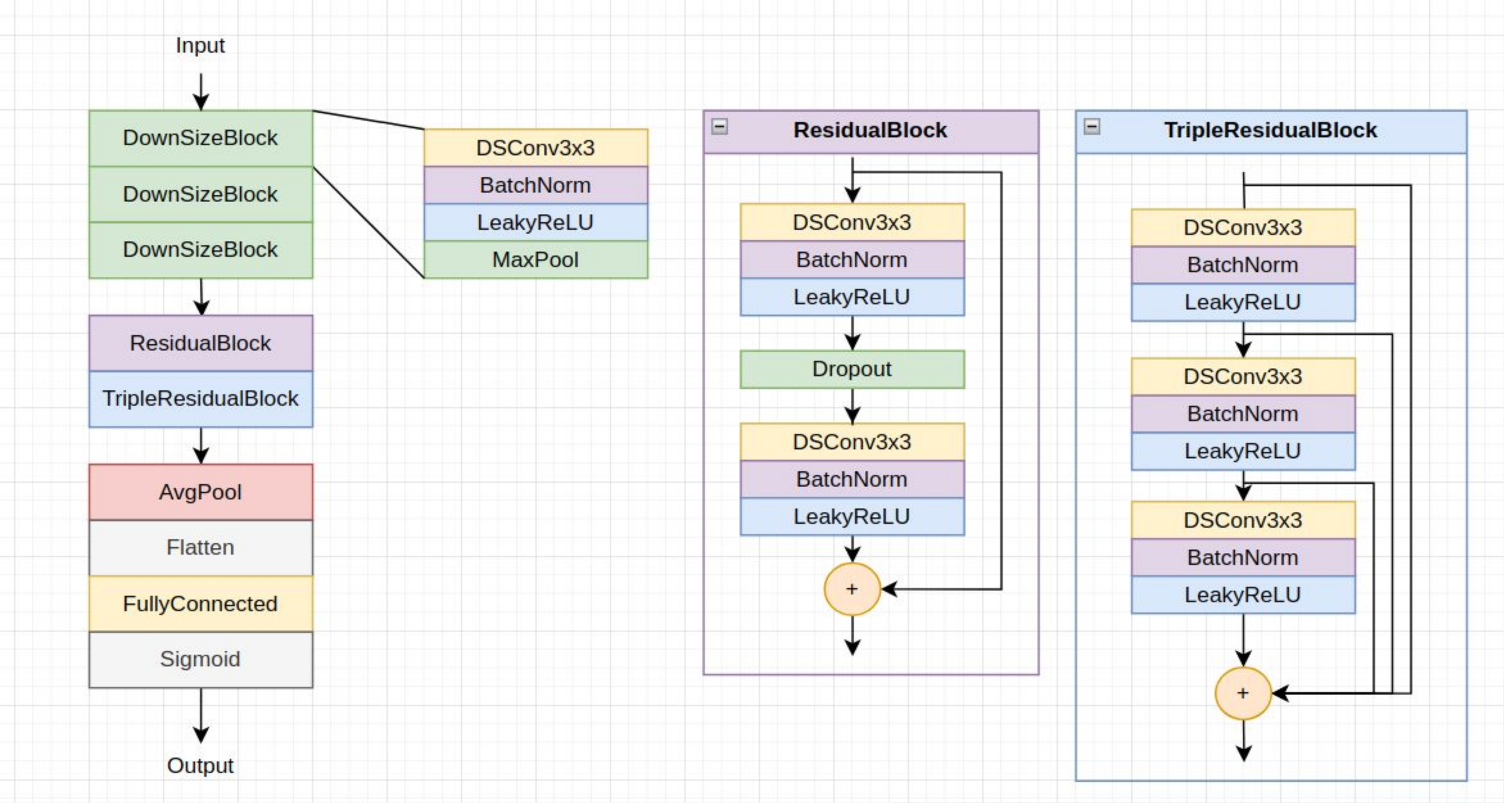
- Reduce computation cost using **average pooling**
- Generate classification result using **fully connected layer**

Model - Scale Choosing

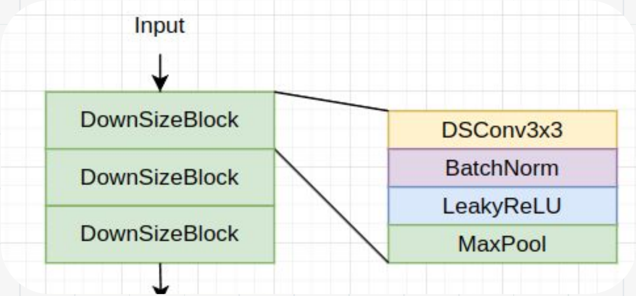


- **Adjust the scale of these 3 part using simple structure**

Model - Structure



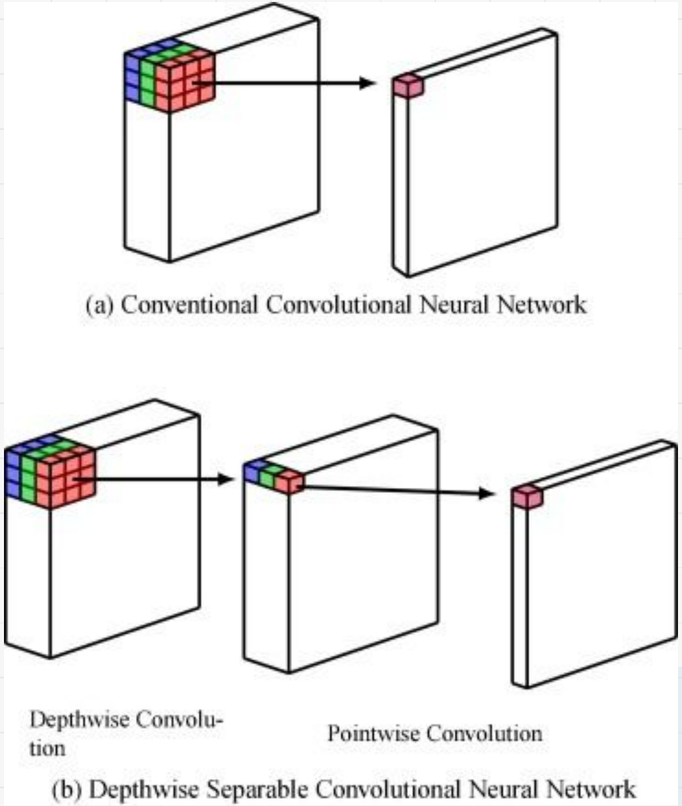
Model - DownSizeBlock



- Depthwise Separable Convolution
- Batch Normalization
- LeakyReLU
- Max Pooling

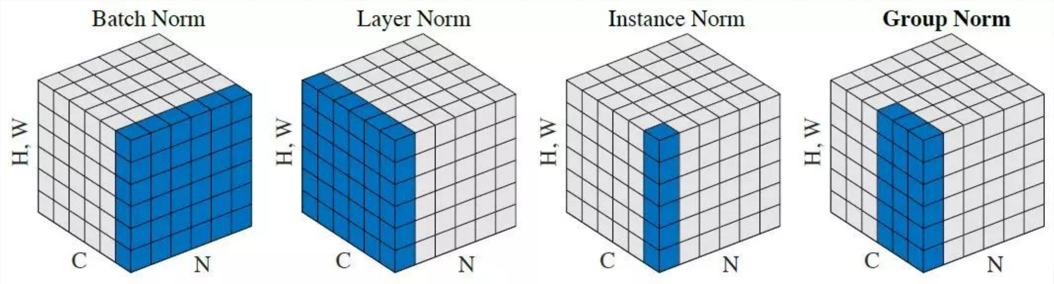
Model - DepthwiseSeperableConv

- Proposed by MobileNet
- Extremely reduce parameter



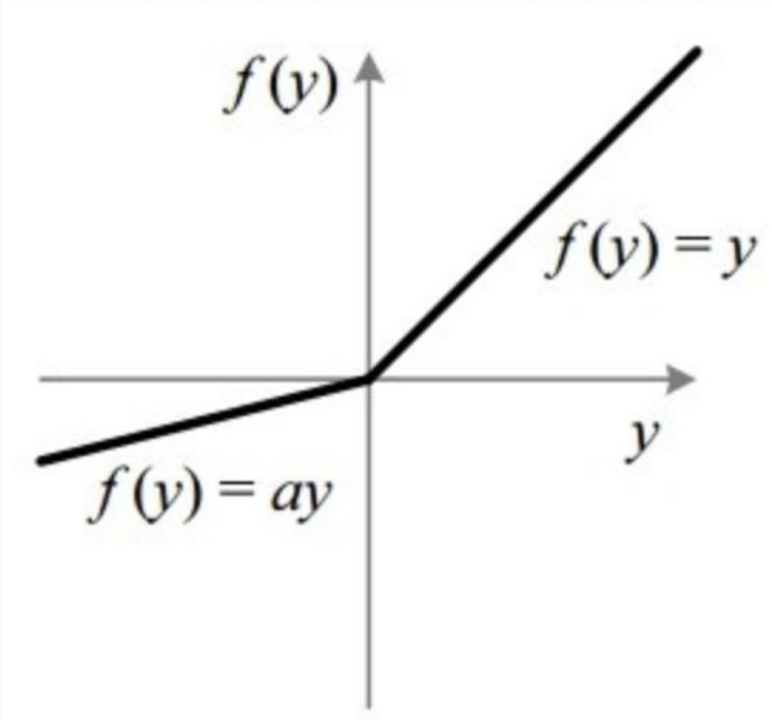
Model - Batch Normalization

- Normalize features among batches
- Increase convergence speed
- Lessen gradient vanishing



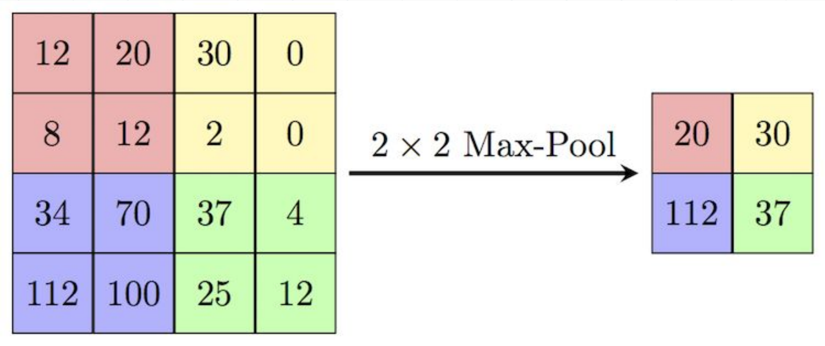
Model - LeakyReLU

- Add non-zero slope on negative part
- Keep information of negative part

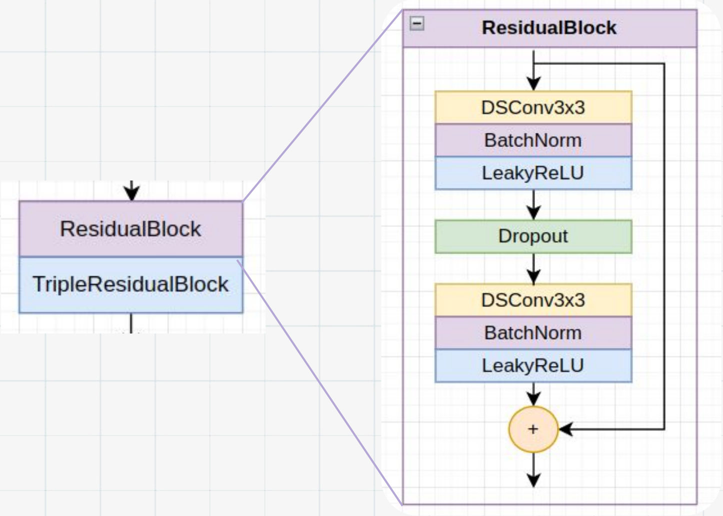


Model - Max Pooling

- Downsize feature maps
- Extract information

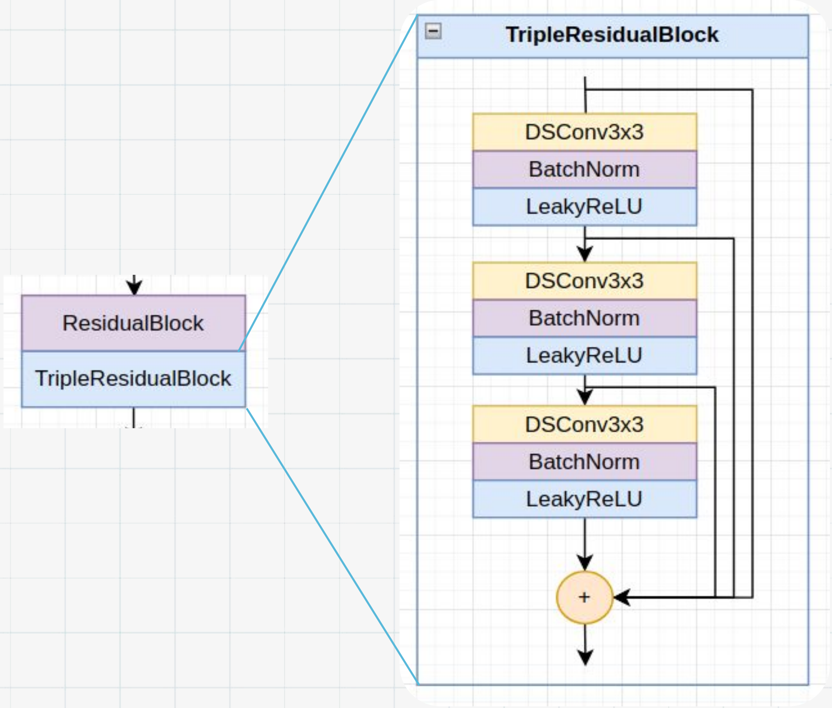


Model - ResidualBlock



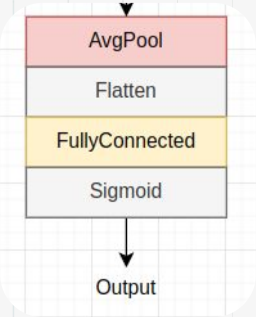
- ResidualBlock act as the first processor, extracting important information from the raw high-dimensional vector.

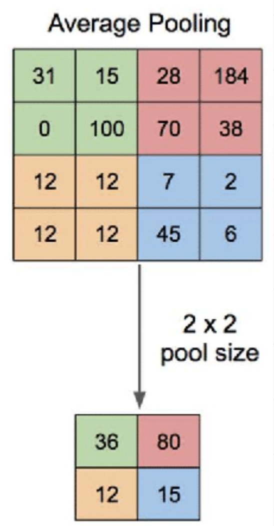
Model - Triple ResidualBlock



- After receiving data from the ResidualBlock, the TripleResidualBlock processes it through three stages. Each stage builds upon the previous one, enhancing the data progressively.

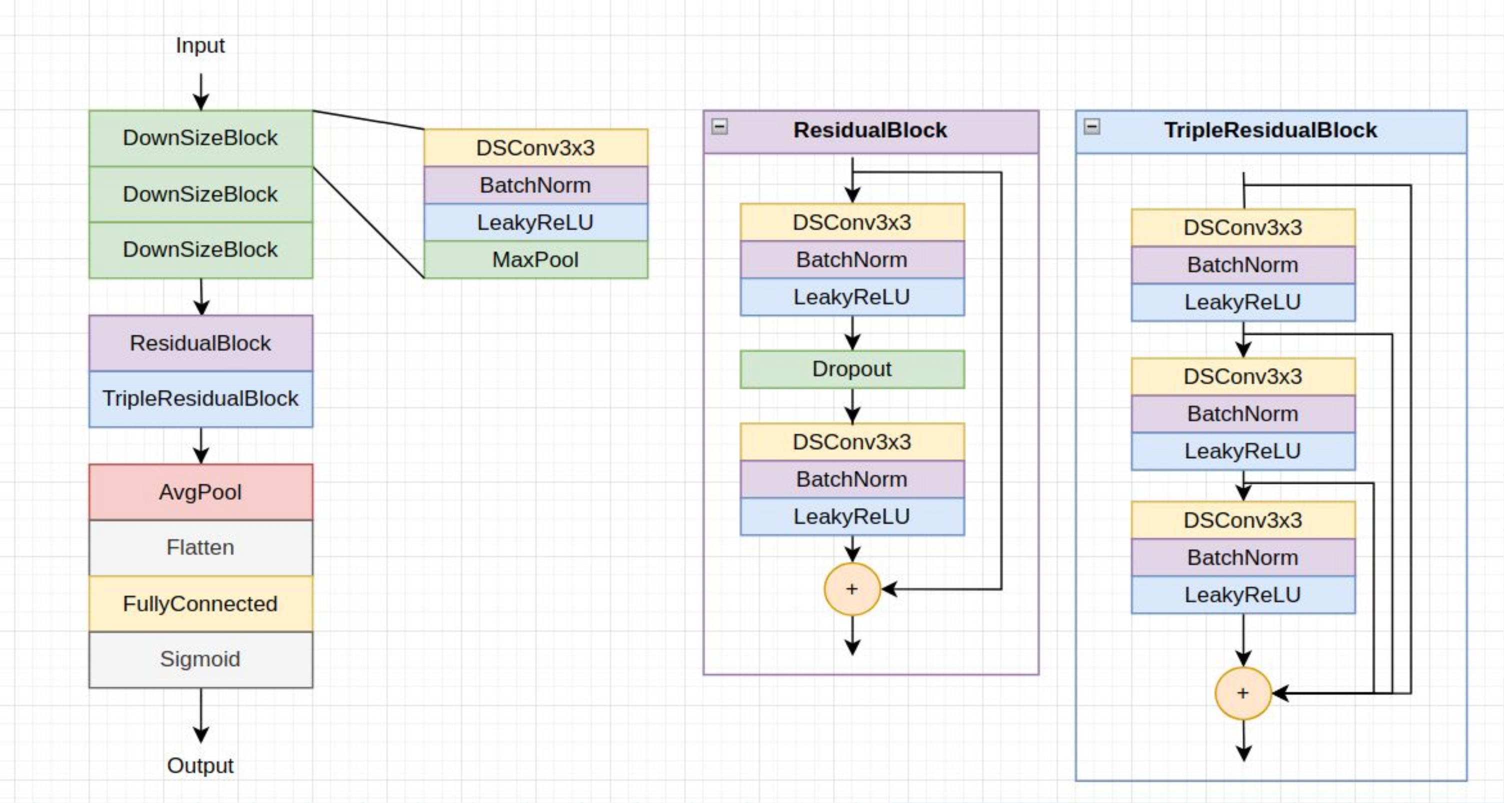
Model - Classifier





- Average Pooling
  - Downsize feature maps
  - **Avoid Large Fully Connected Layer**
- Fully Connected Layer
  - Generate classification result

Model - Structure



Training Method

- Loss function: Binary Cross Entropy
- Optimizer: Adam
  - Gradient Descent
  - Momentum
  - Adaptive Learning Rate
- Batch size = 64, shuffle = True

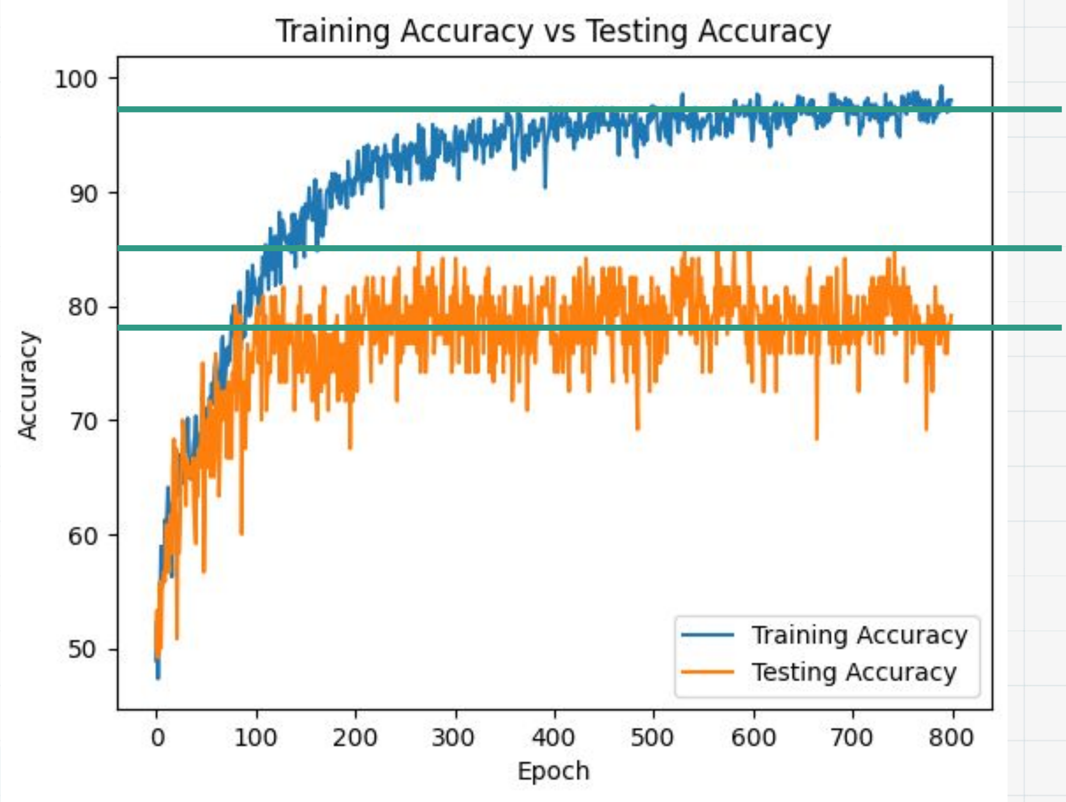
03

Result

Model Size

- Numbers of Parameters: 9.823K
- FLOPS: 17.630M

Learning Curve



Converge at about 97%

85% highest

78% average after 200 epoch

04

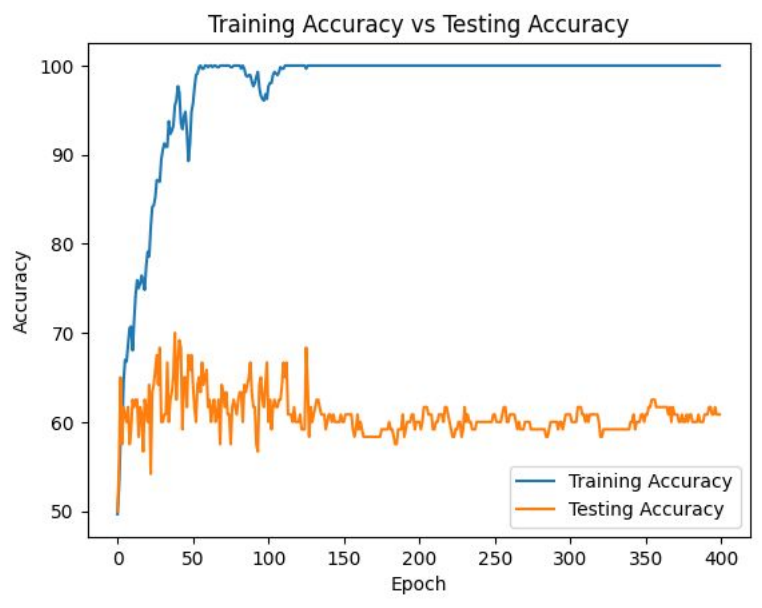
Analysis

Ablation Study

- Data Augmentation
- Activation Function
- Batch Normalization
- Residual network
- Depthwise separable convolution

Data Augmentation

- w/o – 70.0%



- Very fast to converge
- Highly overfitting

Activation Function

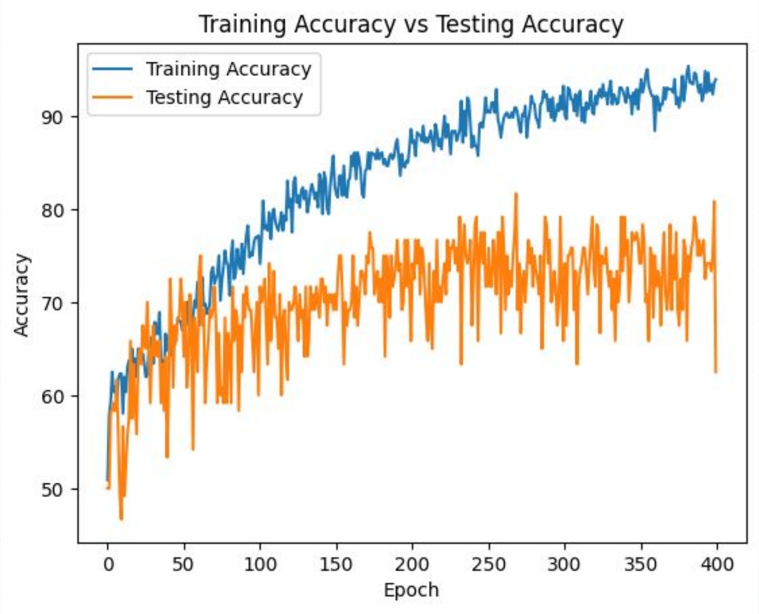
- ReLU – 76.5%
- LeakyReLU – 77.5%
  - Avoids the Dying ReLU Problem
- PReLU – 78%
  - Numbers of parameters: 9.831K
  - FLOPs: 18.564M

Batch Normalization

- BatchNorm – 77.5%
- w/o BatchNorm – 70%
  - Very slow to converge
- GroupNorm – 76.67%

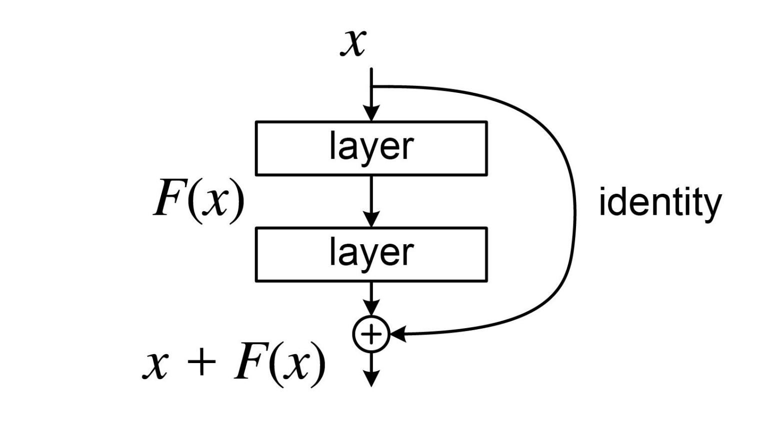
Residual Network

- w/o – 73.5%



Benefits of Residual Connection

- Mitigation of the Vanishing/Exploding Gradient
- Facilitates Training of Deeper Networks

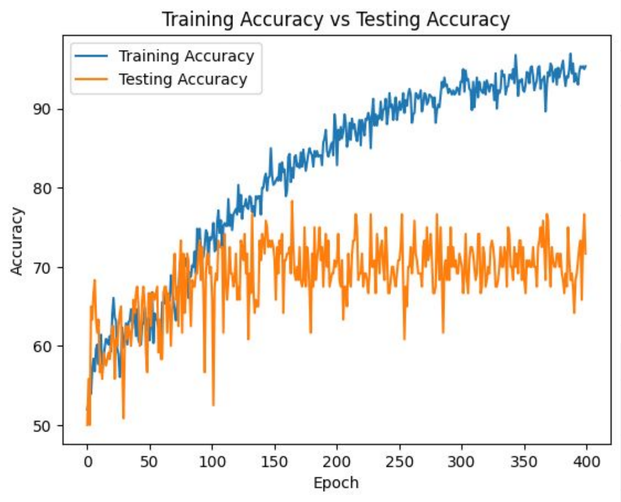


Residual Blocks

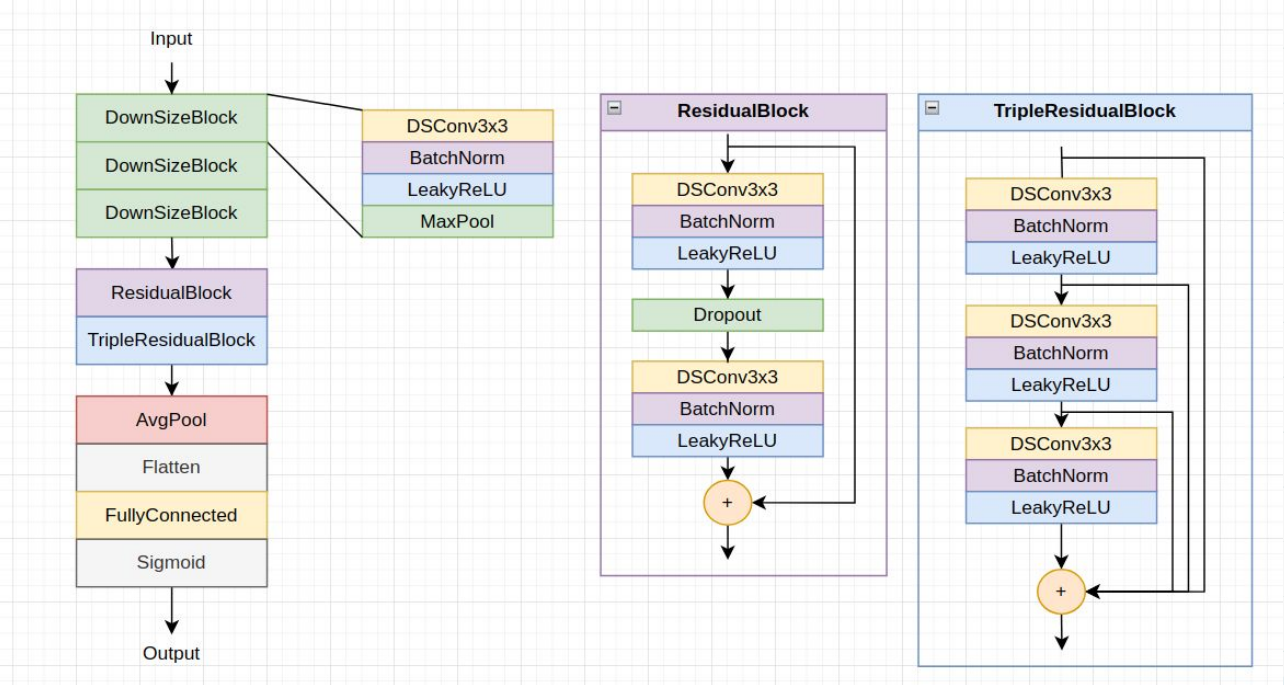
- ResidualBlock + ResidualBlock
- ResidualBlock + TripleResidualBlock
- TripleResidualBlock + ResidualBlock
- TripleResidualBlock + TripleResidualBlock

Depthwise Separable Convolution

- CNN
- Numbers of Parameters = 61.345K
- FLOPs = 72.376M
- Testing Accuracy = 78.5%



Conclusion



- Avg Accuracy: 78%
- Parameters: 9.823K
- FLOPS: 17.630M

Reference

- K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition.
- Howard AG, Zhu M, Chen B, Kalenichenko D, Wang W, Weyand T, et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.