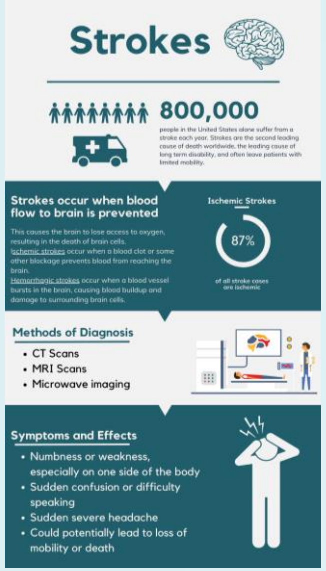
A Machine Learning Method for the Automatic Classification of Strokes

Henry Liu
Advisor: Kevin Crowthers, Ph. D.

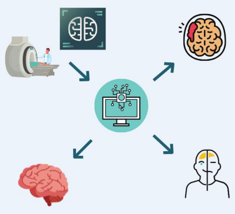
WPI

An Image Classification System Designed to Aid and Accelerate the Stroke Diagnosis and Detection Process

Background



Abstract



Materials

Problem Statement

Current stroke detection methods require too much time for the interpretation of scan results and classification process, delaying treatment and increasing the chance of lasting effects.

Engineering Objective

The objective of this project was to create a machine learning model to automatically detect and classify strokes using CT images in order to aid with the diagnostic process.

Methods

01 — Prelabeled CT images from various medical centers were collected

02 — Each image was normalized and used to create a training, testing, and validation set

03 — Using the images from the training and validation sets, an image classification model was developed

04 — The model was evaluated using the testing data, and the model's metrics were collected

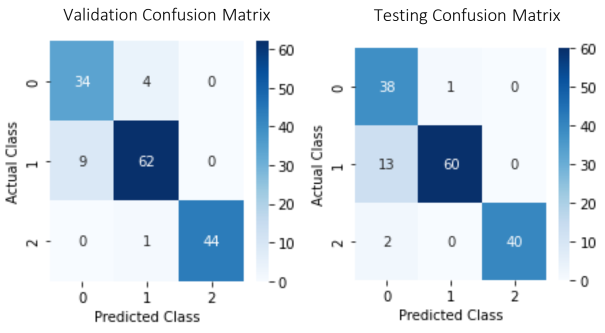
05 — Steps 3 and 4 were repeated until the optimal model was developed

Design Matrix

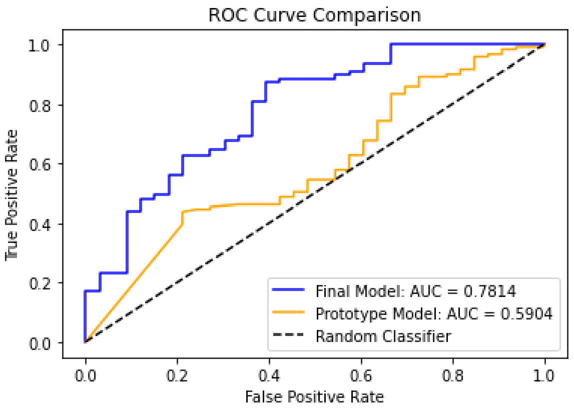
Evaluation of Common Classification Techniques in Image Classification

Criteria	Max	Decision Tree	SVM	KNN	CNN
Suitable with small datasets	**6**	3	5	2	4
High accuracy of output	**10**	6	8	7	8
High testing and training speed	**9**	6	8	4	8
Capable of dealing with large number of features	**8**	3	5	6	7
Easily improved and configured	**7**	3	5	5	6
Total	**40**	21	31	24	33
Percent	**100%**	53%	78%	60%	83%

Results



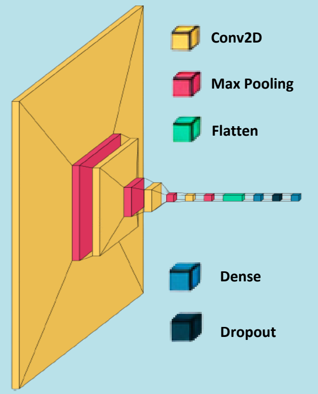
0 – Normal Scans | 1 – Hemorrhagic Strokes | 2 – Ischemic Strokes



Conclusions

- Final model classifies strokes with **89.6%** testing accuracy and **90.3%** validation accuracy
- Demonstrates the use of neural networks as a viable method of stroke classification
- Final model returns chance of abnormality from cranial CT scan
- Final model was a significant improvement over the first model, with a p-value of **0.0001**

Model Architecture



Extensions

- Testing on larger datasets
- Increasing classes to include other cranial diseases
- Testing during the real-time stroke diagnosis process
- Testing on other types of medical imaging (MRI or microwave imaging)

References

1. Chawla, M., Sharma, S., Sivaswamy, J., & Kishore, L. (2009). A method for automatic detection and classification of stroke from brain CT images. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society.* doi:10.1109/iembs.2009.5335289

2. Ireland, D., & Bialkowski, M. E. (2011). Microwave Head Imaging for Stroke Detection. *Progress In Electromagnetics Research M*, 21, 163-175. doi:10.2528/pierm11082907

A Machine Learning Method for the Automatic Classification of Strokes

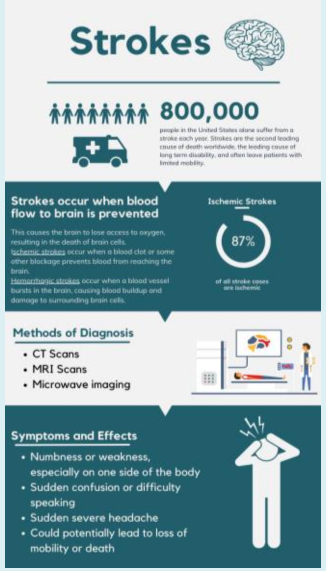
A Machine Learning Method for the Automatic Classification of Strokes

Henry Liu
Advisor: Kevin Crowthers, Ph. D.

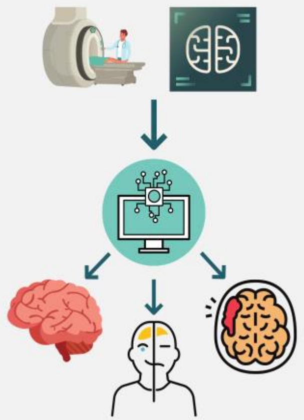
WPI

An Image Classification Model Designed to Aid and Accelerate the Stroke Diagnosis and Detection Process

Background



Abstract



Problem Statement

Current stroke detection methods require too much time for the interpretation of scan results and classification process, delaying treatment and increasing the chance of lasting effects.

Engineering Objective

The objective of this project was to create a machine learning model to automatically detect and classify strokes using CT images in order to aid with the diagnostic process.

Methods

01 — Prelabeled CT images from various medical centers were collected

02 — Each image was normalized and used to create a training, testing, and validation set

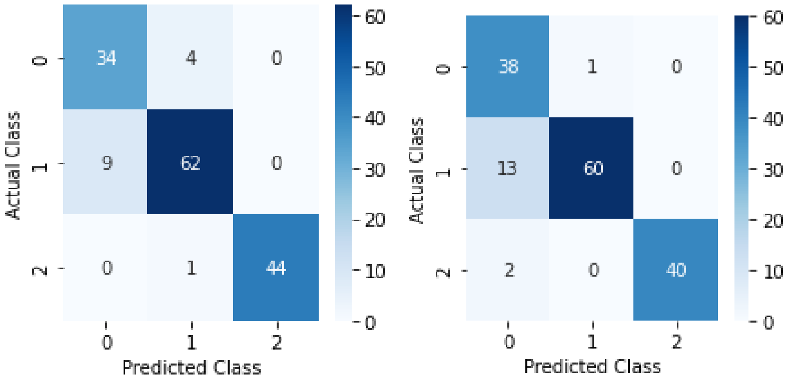
03 — Using the images from the training and validation sets, an image classification model was developed

04 — The model was evaluated using the testing data, and the model's metrics were collected

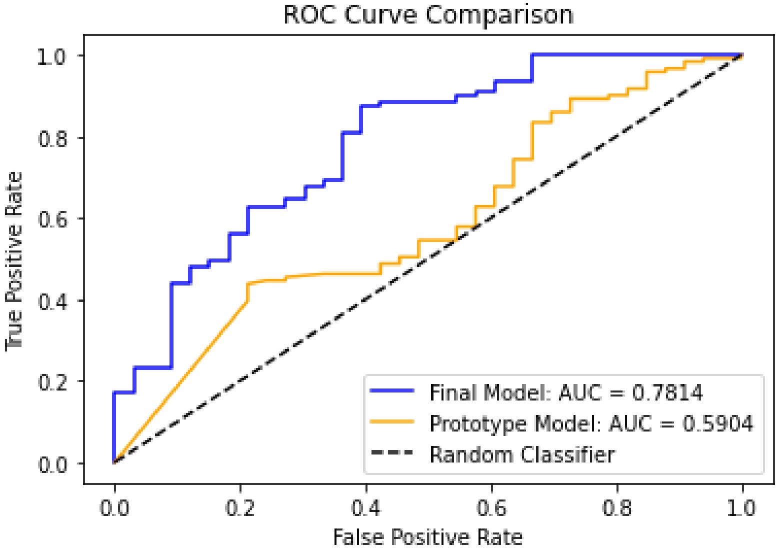
05 — Steps 3 and 4 were repeated until the optimal model was developed

Materials

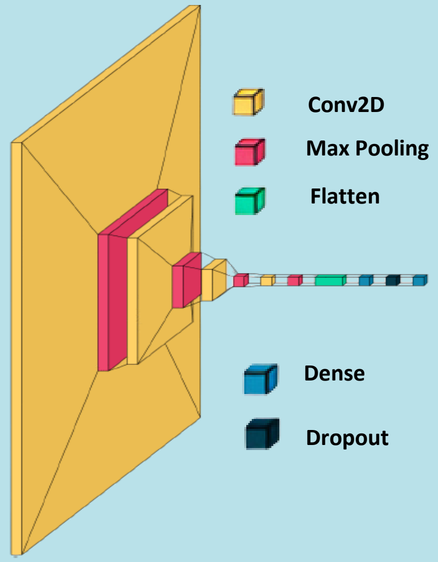
Results



0 – Normal Scans | 1 – Hemorrhagic Strokes | 2 – Ischemic Strokes



Model Architecture



Conclusions

- Final model classifies strokes with high accuracy
- **89.6%** testing accuracy
- **90.3%** validation accuracy
- Demonstrates the use of neural networks as a viable method of stroke classification
- Final model able to return chance of abnormality from cranial CT scan

Extensions

- Testing on larger datasets
- Increasing classes to include other cranial diseases
- Testing during the real-time stroke diagnosis process
- Testing on other types of medical imaging (MRI or microwave imaging)

Contents

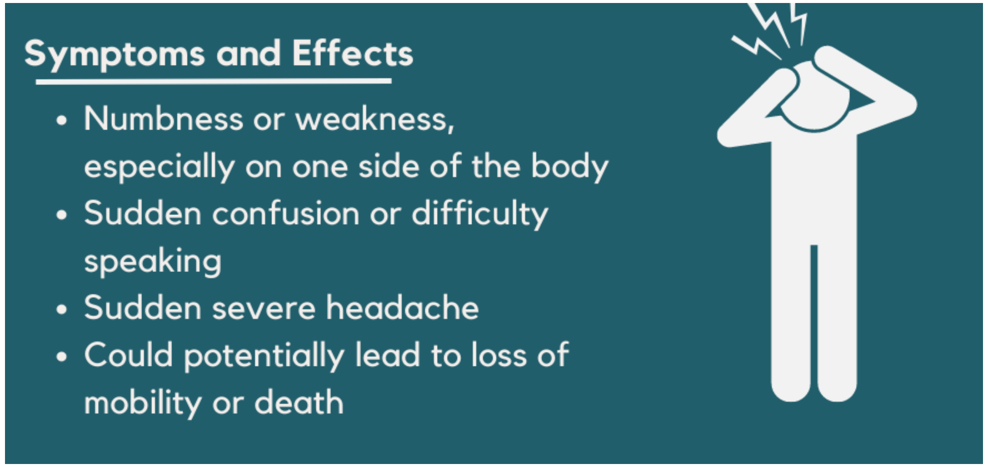
1.

Background and Objectives

Strokes
Stroke Detection
Image Classification
Project Objectives

Strokes – General Information

- Ischemic strokes – blood clot prevents blood from reaching the brain
- Hemorrhagic strokes – blood vessel bursts in the brain and causes blood buildup



Stroke Detection – Diagnostic Process

- Begins with physical examination by physician
- Medical Imaging
  - CT Scans
  - MRI Scans
  - Microwave Imaging
- Examination, interpretation, and classification

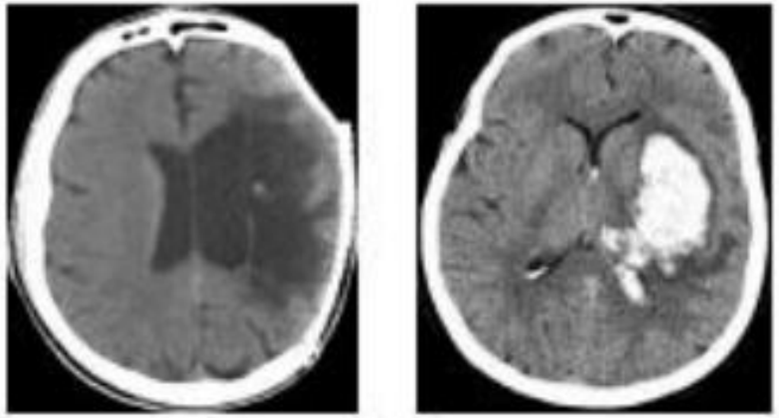
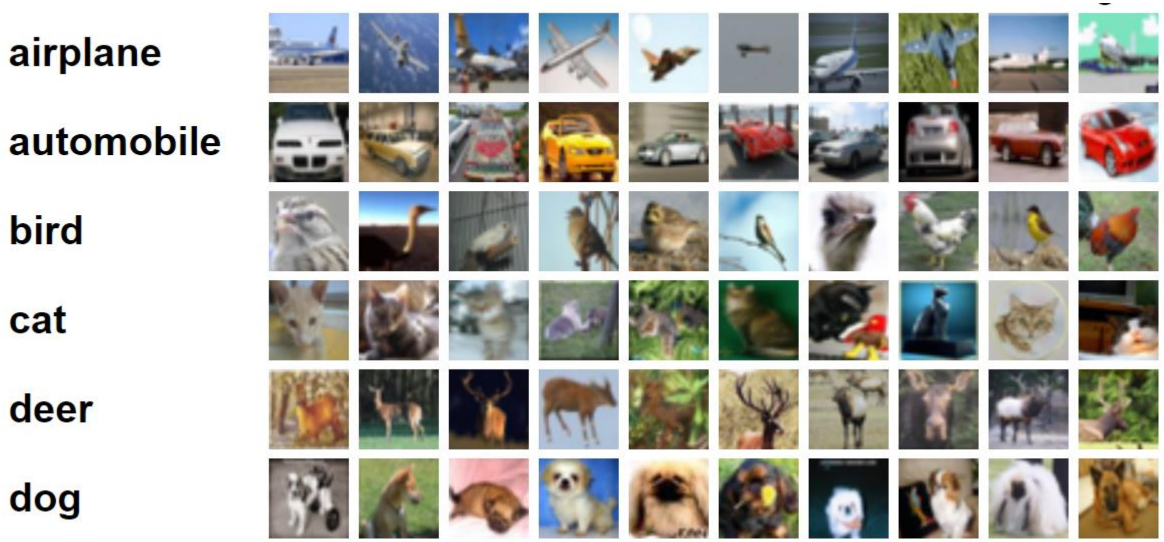


Image Classification

- Branch of machine learning
- Supervised learning
- Models assign labels to images using patterns determined from training data



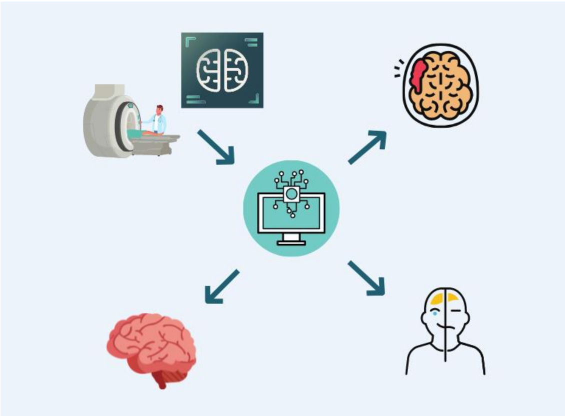
Problem and Objective

Problem Statement

- The current stroke diagnostic process is hindered by the need for human interpretation and classification of medical scans
- Requires too much time and delays treatment to the patient

Engineering Objective

- The objective of this project is to design an image classification model to automatically detect and classify strokes using images from CT scans



2.

Materials and Methods

Design Criteria and Matrix
Materials Used
Procedures

Design Criteria

- Be able to evaluate inputs with high number of features
- Must have high accuracy outputs
- Must have low testing and training times
- Should be suitable with small datasets (~500-1000 images)
- Should be able to be configured and improved easily

Evaluation of Common Classification Techniques in Image Classification					
Criteria	**Max**	**Decision Tree**	**SVM**	**KNN**	**CNN**
Suitable with small datasets	**6**	3	5	2	4
High accuracy of output	**10**	6	8	7	8
High testing and training speed	**9**	6	8	4	8
Capable of dealing with large number of features	**8**	3	5	6	7
Easily improved and configured	**7**	3	5	5	6
Total	**40**	21	31	24	33
Percent	**100%**	53%	78%	60%	83%

Decision Matrix

Comparing common classification algorithms and their applications in image classification

Materials Used

- **Python** (programming language)
- **Google Colaboratory** (Jupyter Notebook runtime environment)
- **Keras** (deep-learning API for Python)
- Datasets of CT scans of strokes patients (both ischemic and hemorrhagic) and healthy patients

01

Prelabeled CT images from various medical centers were collected

02

Each image was normalized and used to create a training, testing, and validation set

Image Collection and Processing

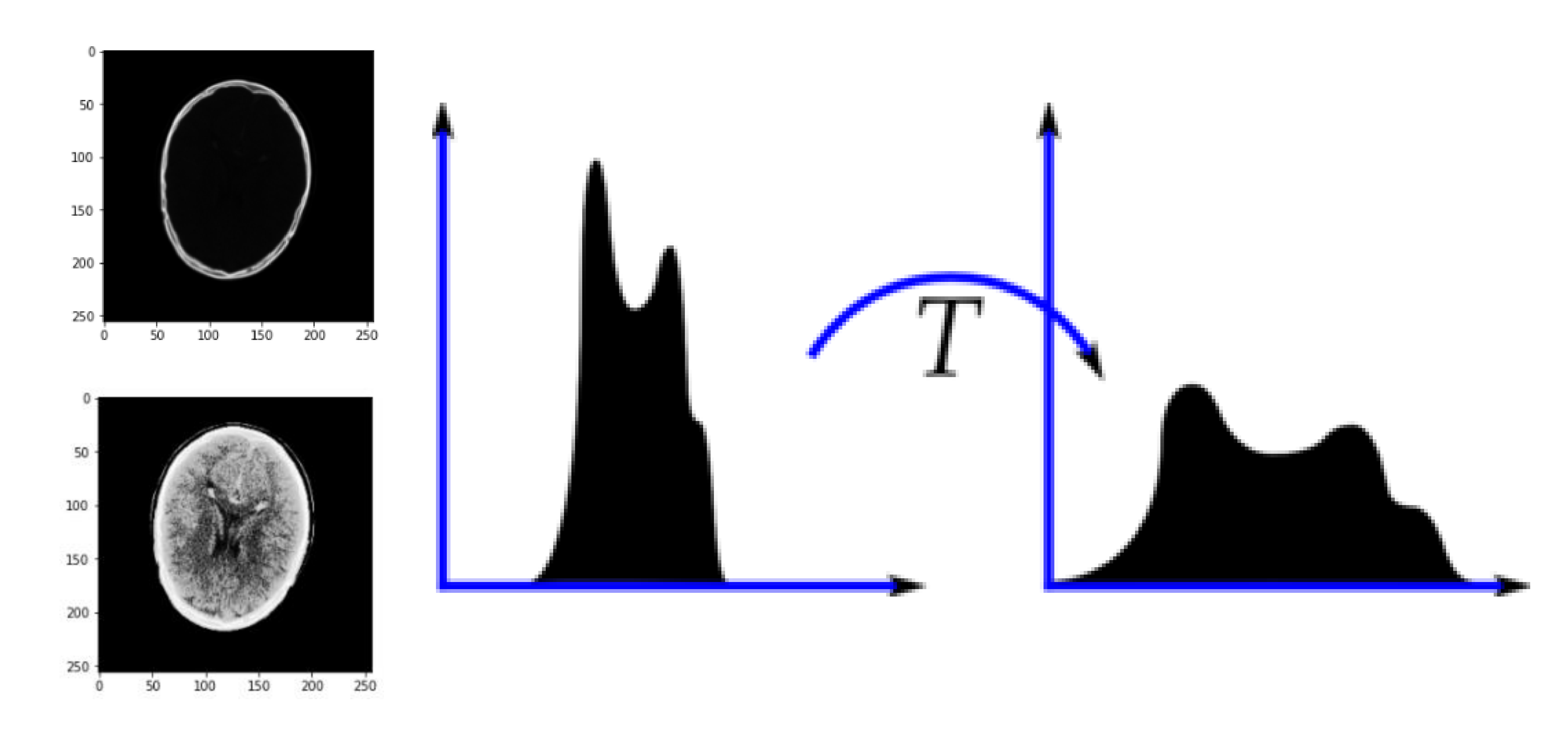


Image Equalization

03 — Using the images from the training and validation sets, an image classification model was developed

04 — The model was evaluated using the testing data, and the model's metrics were collected

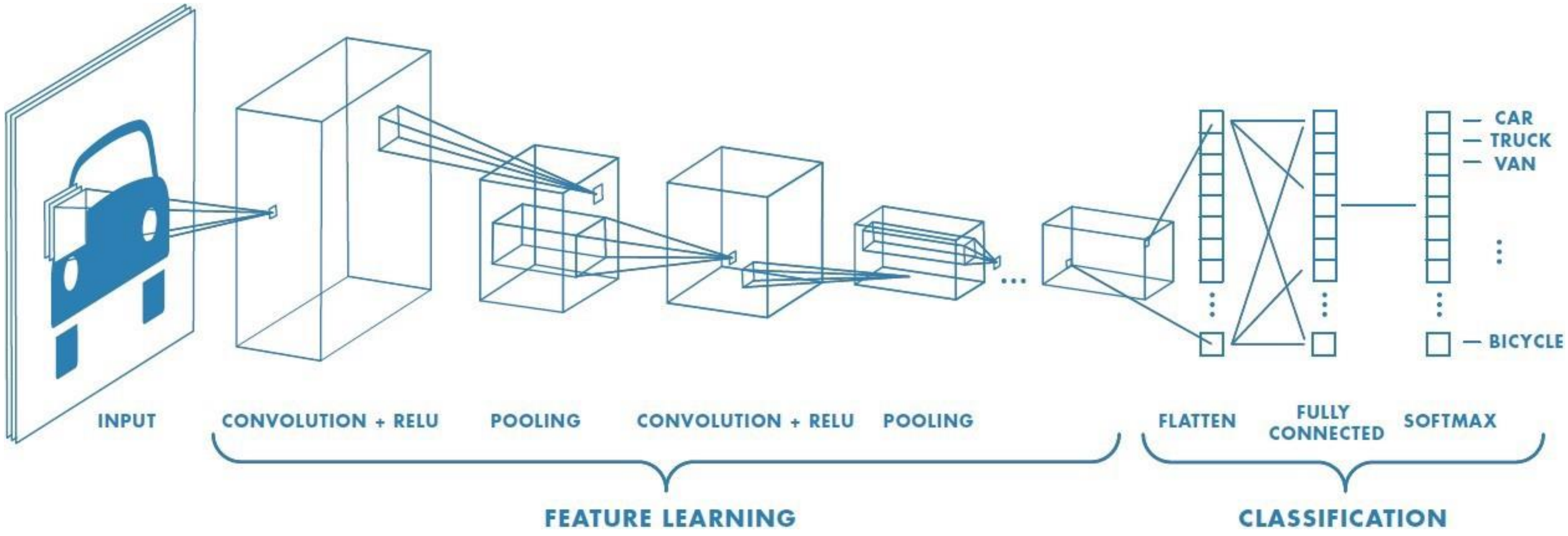
05 — Steps 3 and 4 were repeated until the optimal model was developed

Model Creation and Testing

3.

Prototypes and Designs

Prototype Models
Primary Models
Final Model



Example Convolutional Neural Network Structure

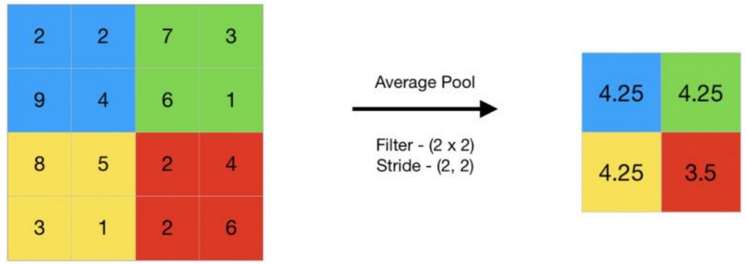
Prototypes

Purpose and Structure

- Binary classification models
- Used to find effective designs for later variations
- Composed of 3 convolutional layers (64 filters each, Average Pooling)

Results

- Much overfitting (often a result of too many parameters or layers)
- Provided model to be used as baseline for primary models



Primary Models - Architecture

Primary Model 1

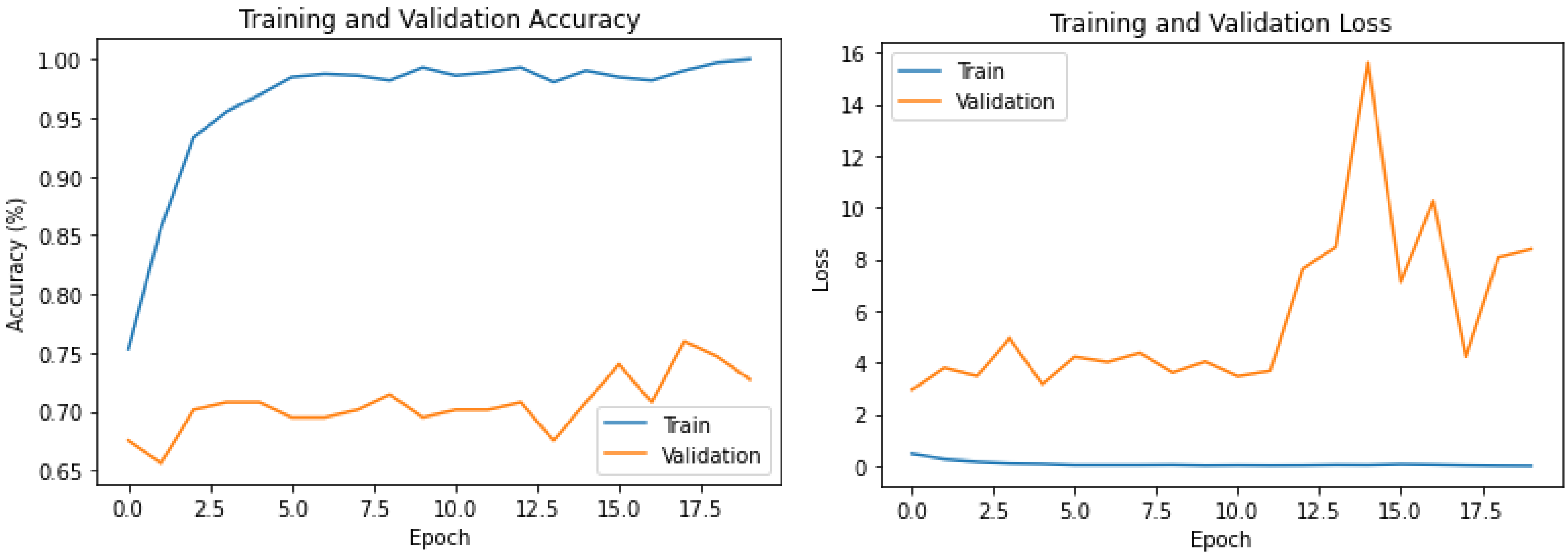
- 3 convolutional layers
  - Kernel sizes: 9x9, 5x5, 3x3
  - Filters: 32, 64, 128
- 2x2 Average Pooling layers following convolutional layers
- Flatten layer, dense layer (64 output), dense layer (final output)

Primary Model 2

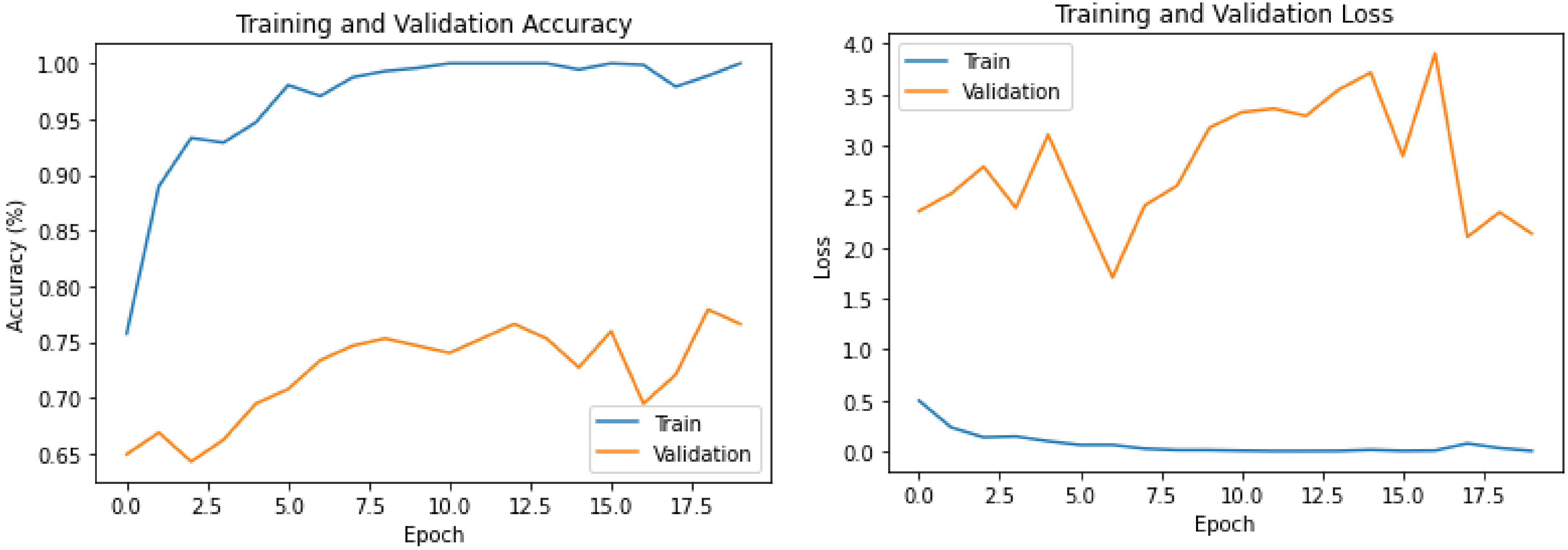
- 3 convolutional layers
  - Kernel sizes: 9x9, 5x5, 3x3
  - Filters: 16, 32, 64
- 2x2 **Max Pooling** layers following convolutional layers
- Flatten layer, dense layer (64 output), dense layer (final output)

Final Model

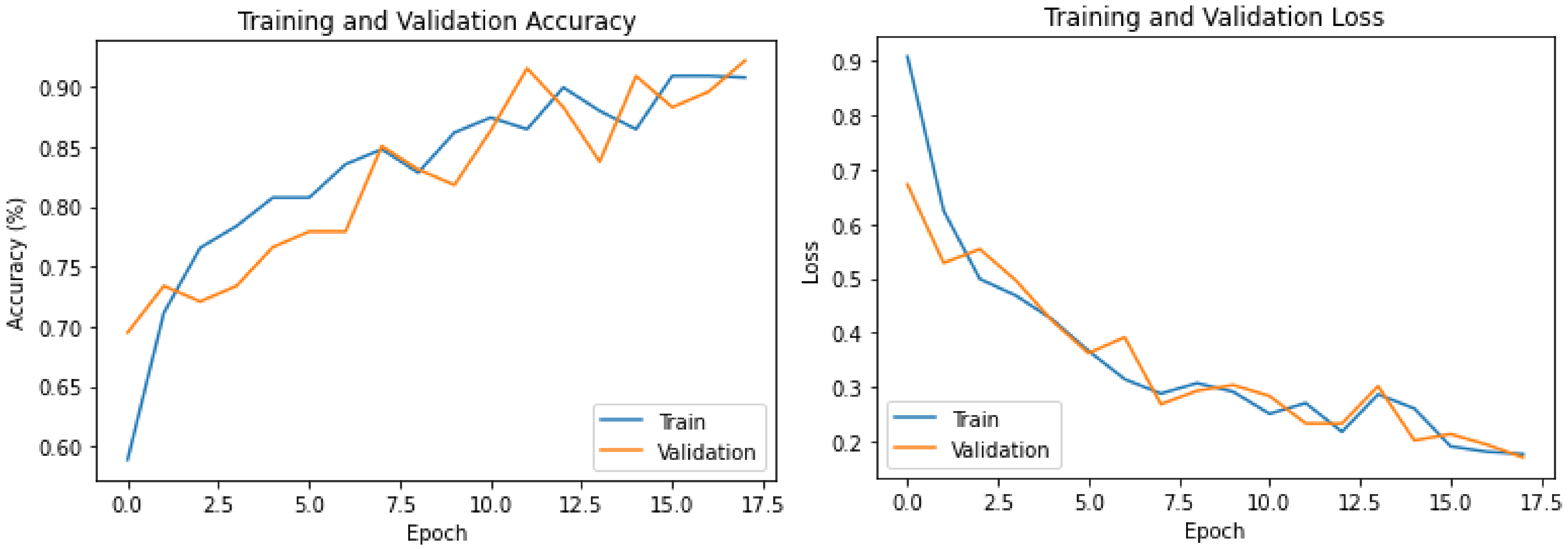
- 4 convolutional layers
  - Kernel sizes: 9x9, 7x7, 7x7, 3x3
  - Filters: 32, 64, 64, 128
- 3x3 or 2x2 Max Pooling layers following convolutional layers
- Flatten layer, dense layer (64 output), dense layer (final output)
- **50% dropout layer** following first dense layer



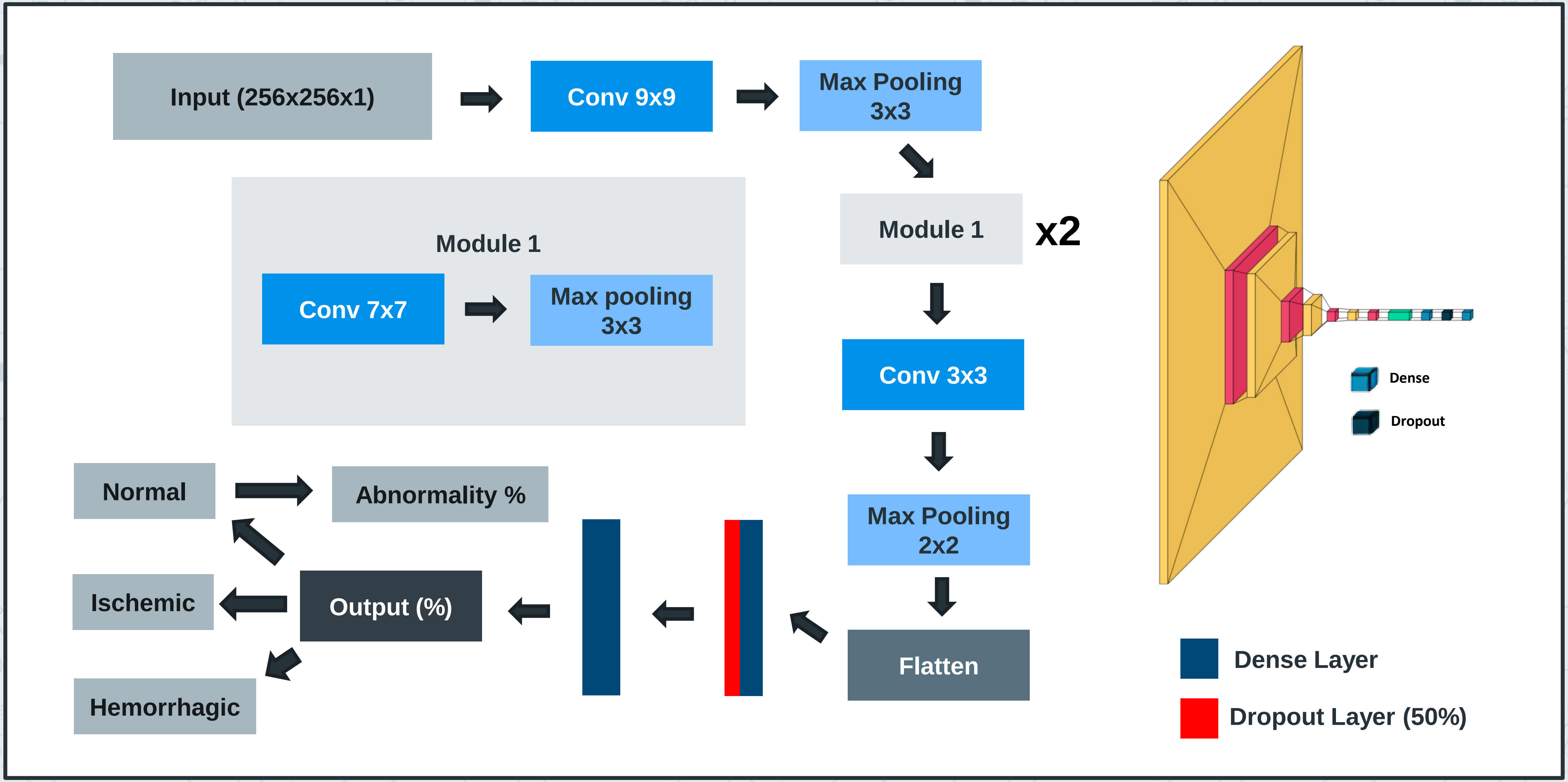
Primary Model 1 – Prototype Design



Primary Model 2 – Introduction of Max Pooling



Primary Model 3 – Introduction of Dropout Layer



Final Model – Architecture

4.

Results

Model Metrics
Final Model Results

Model Comparison (Metrics)

Primary Model 1

- Accuracy: 76.6%
- Loss: 7.839
- Runtime: 53ms per batch

- 3,085,315 parameters

- Severe overfitting to training data

Primary Model 2

- Accuracy: 79.2%
- Loss: 2.239
- Runtime: 26ms per batch

- 557,443 parameters

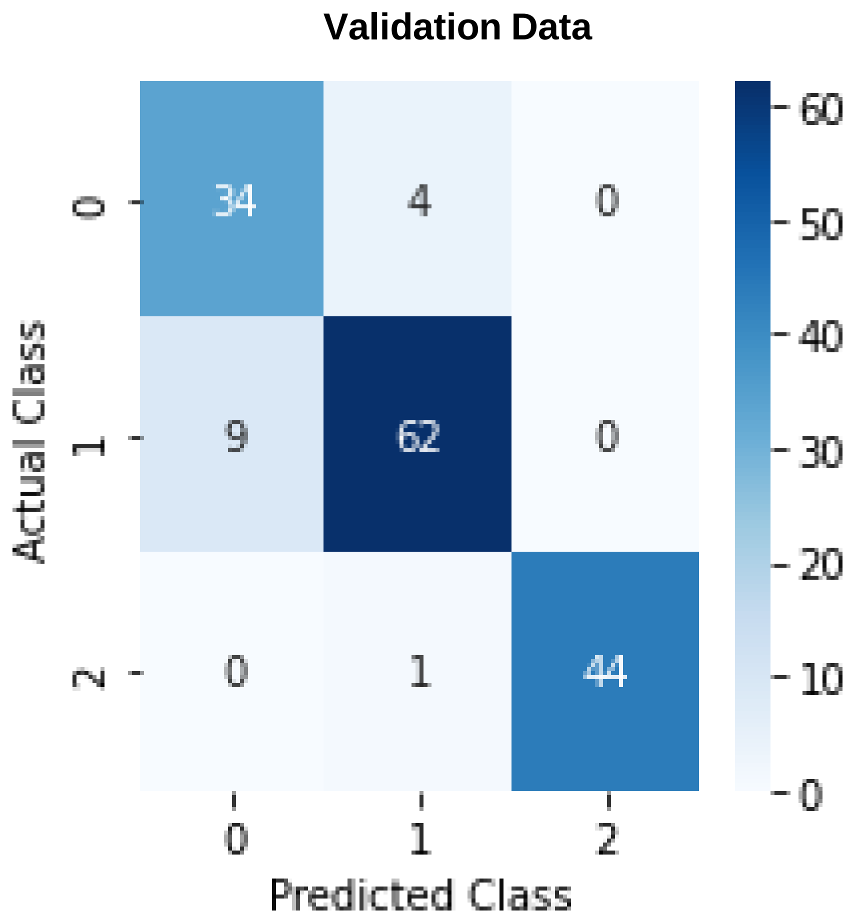
- Moderate overfitting to data

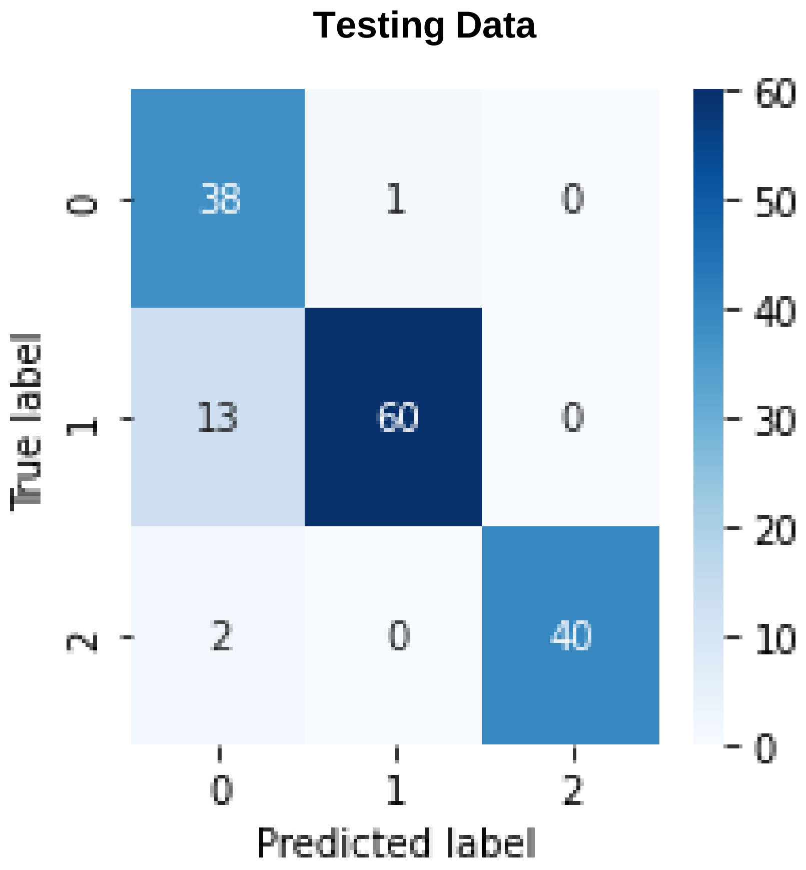
Final Model

- Accuracy: 89.6%
- Loss: 0.223
- Runtime: 19ms per batch

- 410,691 parameters

- Highest accuracy and lowest loss values





Final Model – Confusion Matrices

5.

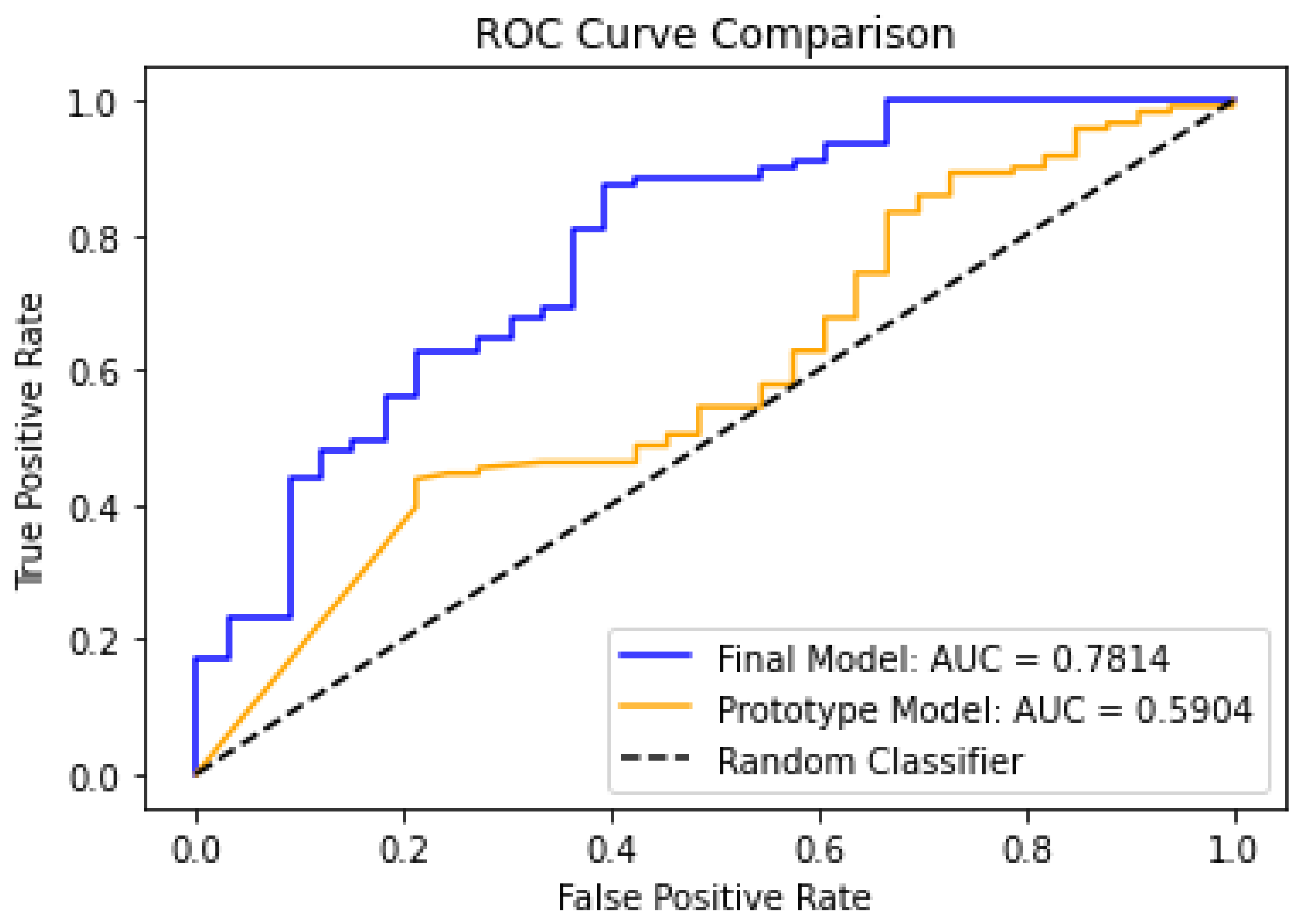
Analysis, Conclusion, and Future Extensions

Result Analysis and Significance
Conclusions
Future Possibilities

Model Comparison – McNemar's Test

		Model 1		
		Correct	Incorrect	Total
Model 3	Correct	114	25	139
	Incorrect	4	11	15
	Total	118	36	154

$\chi2$	15.2069
p-value	0.0001



Model Analysis – ROC Curve and Precision-Recall

Conclusion

- Final model classifies with high accuracy
- **89.6%** testing accuracy and **90.3%** validation accuracy
- Demonstrates the use of neural networks as a viable method of stroke classification
- Final model able to return chance of abnormality from cranial CT scan
- Final model was a significant improve over the primary model 1, with a p-value of ~0.0001

Future Extensions

- Testing on larger datasets
- Increasing classes to include other diseases
- Testing during the real-time stroke diagnosis process
- Testing on other types of medical imaging

References

About Stroke. (2021, August 02). Retrieved from https://www.cdc.gov/stroke/about.htm

Chawla, M., Sharma, S., Sivaswamy, J., & Kishore, L. (2009). A method for automatic detection and classification of stroke from brain CT images. *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*. doi:10.1109/iembs.2009.5335289

Chilamkurthy, S., Ghosh, R., Tanamala, S., Biviji, M., Campeau, N. G., Venugopal, V. K., Mahajan, V., Rao, P., & Warier, P. (2018, March 13). CQ500 [Dataset]. Qure.ai. http://headctstudy.qure.ai/dataset

CT scan. (2020, February 28). Retrieved from https://www.mayoclinic.org/tests-procedures/ct-scan/about/pac-20393675 #:~:text=A computerized tomography (CT) scan

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. Array programming with NumPy. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2

Ireland, D., & Bialkowski, M. E. (2011). Microwave Head Imaging for Stroke Detection. *Progress In Electromagnetics Research M*, *21*, 163-175. doi:10.2528/pierm11082907

Kanellopoulos, I., & Wilkinson, G. G. (1997). Strategies and best practice for neural network image classification. *International Journal of Remote Sensing*, *18*(4), *711-725*.

Kistler et al. The virtual skeleton database: an open access repository for biomedical research and collaboration. *JMIR*, 2013 http://doi.org//10.2196/jmir.2930

Oskar Maier et al. ISLES 2015 - A public evaluation benchmark for ischemic stroke lesion segmentation from multispectral MRI Medical Image Analysis, Available online 21 July 2016, ISSN 1361-8415 http://dx.doi.org/10.1016/j.media.2016.07.009.

Rahman, A. (2021, June 18). Brain Stroke CT Image Dataset (Version 1) [Dataset]. Afridi Rahman. https://www.kaggle.com/afridirahman/brain-stroke-ct-image-dataset

Soofi, A., & Awan, A. (2017). Classification Techniques in Machine Learning: Applications and Issues. *Journal of Basic & Applied Sciences*, *13*, 459-465. doi:10.6000/1927-5129.2017.13.76

Timeline of a Stroke. (2017, November 21). WebMD. https://www.webmd.com/stroke/stroke-symptoms-timeline

A Machine Learning Method for the Automatic Classification of Strokes

Henry Liu
Advisor: Kevin Crowthers, Ph. D.
WPI

An Image Classification Model Designed to Aid and Accelerate the Stroke Diagnosis and Detection Process

Background

Strokes

800,000 people in the United States alone suffer from a stroke each year. Strokes are the second leading cause of death worldwide, the leading cause of long term disability, and often leave patients with limited mobility.

Strokes occur when blood flow to brain is prevented

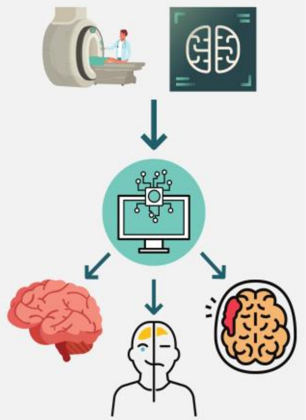
This causes the brain to lose access to oxygen, resulting in the death of brain cells.
Ischemic strokes occur when a blood clot or some other blockage prevents blood from reaching the brain.
Hemorrhagic strokes occur when a blood vessel bursts in the brain, causing blood buildup and damage to surrounding brain cells.

Ischemic Strokes
87% of all stroke cases are ischemic

Methods of Diagnosis
- CT Scans
- MRI Scans
- Microwave imaging

Symptoms and Effects
- Numbness or weakness, especially on one side of the body
- Sudden confusion or difficulty speaking
- Sudden severe headache
- Could potentially lead to loss of mobility or death

Abstract



Problem Statement

Current stroke detection methods require too much time for the interpretation of scan results and classification process, delaying treatment and increasing the chance of lasting effects.

Engineering Objective

The objective of this project was to create a machine learning model to automatically detect and classify strokes using CT images in order to aid with the diagnostic process.

Methods

01 — Prelabeled CT images from various medical centers were collected

02 — Each image was normalized and used to create a training, testing, and validation set

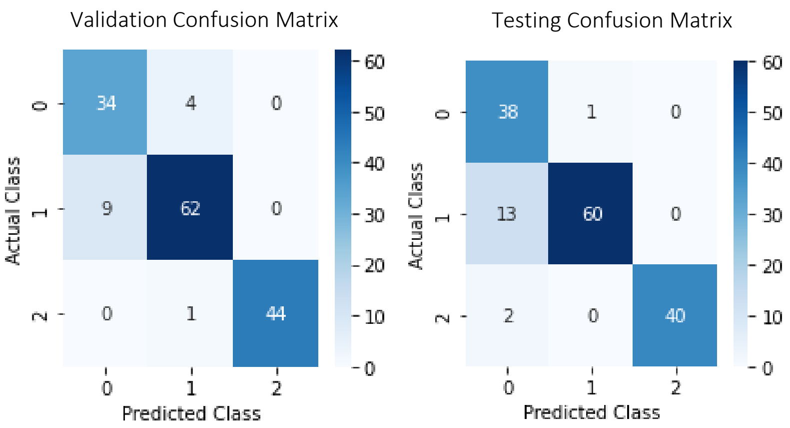
03 — Using the images from the training and validation sets, an image classification model was developed

04 — The model was evaluated using the testing data, and the model's metrics were collected

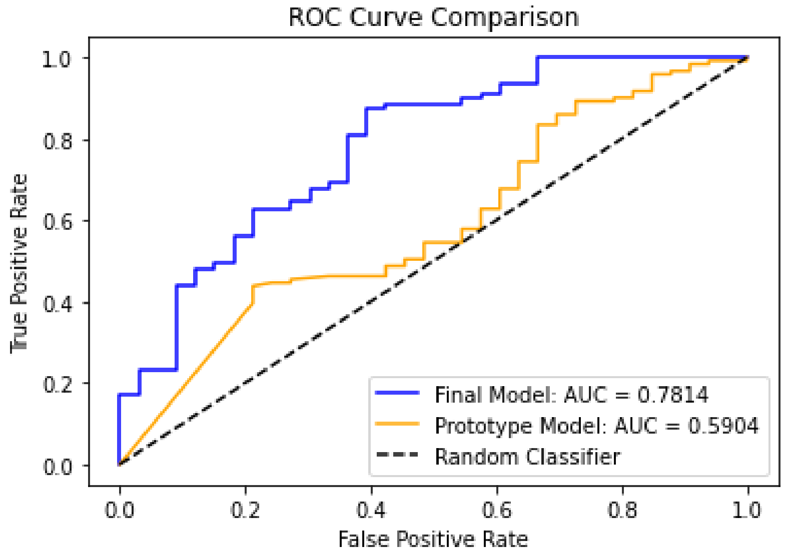
05 — Steps 3 and 4 were repeated until the optimal model was developed

Materials

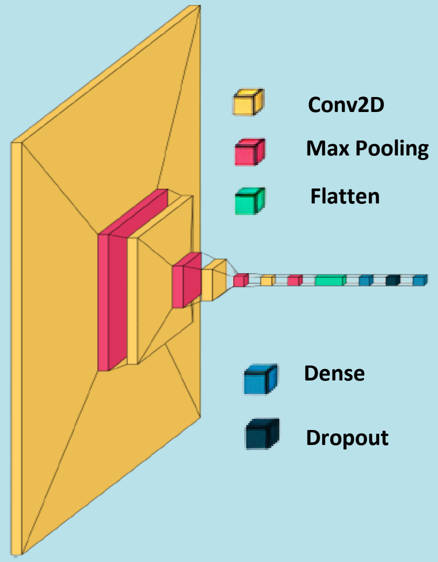
Results



0 – Normal Scans | 1 – Hemorrhagic Strokes | 2 – Ischemic Strokes



Model Architecture



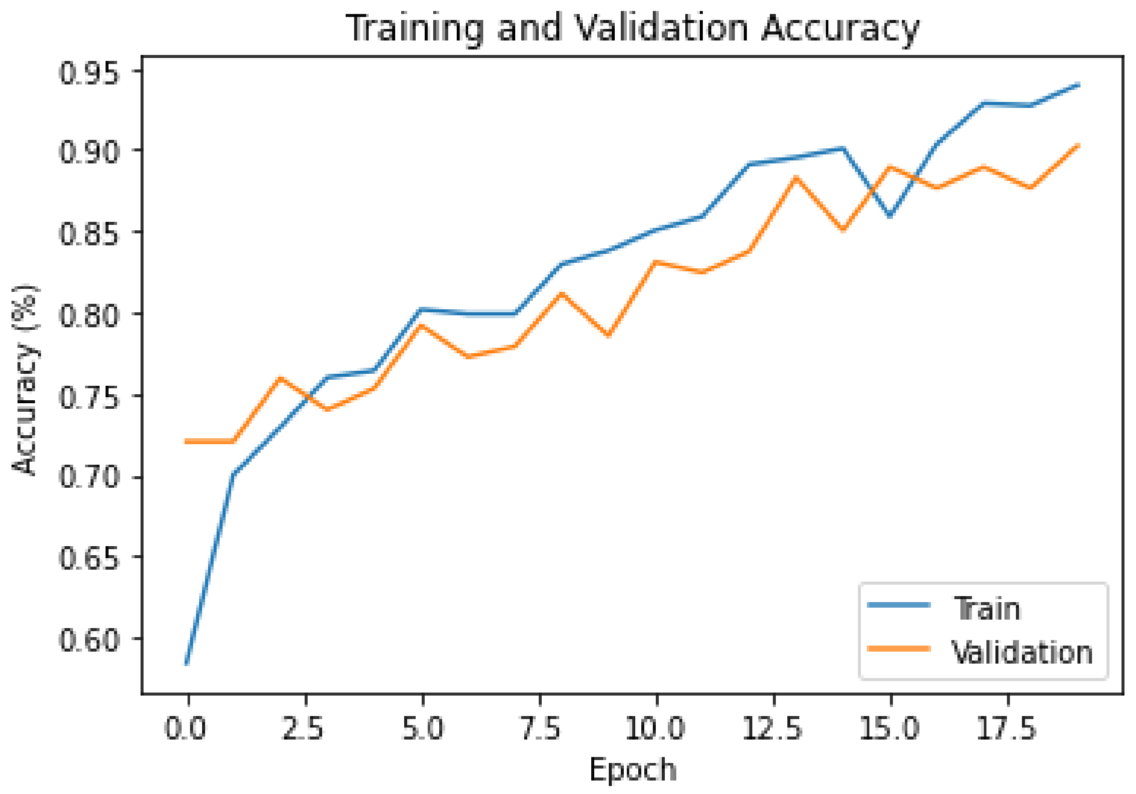
Conclusions

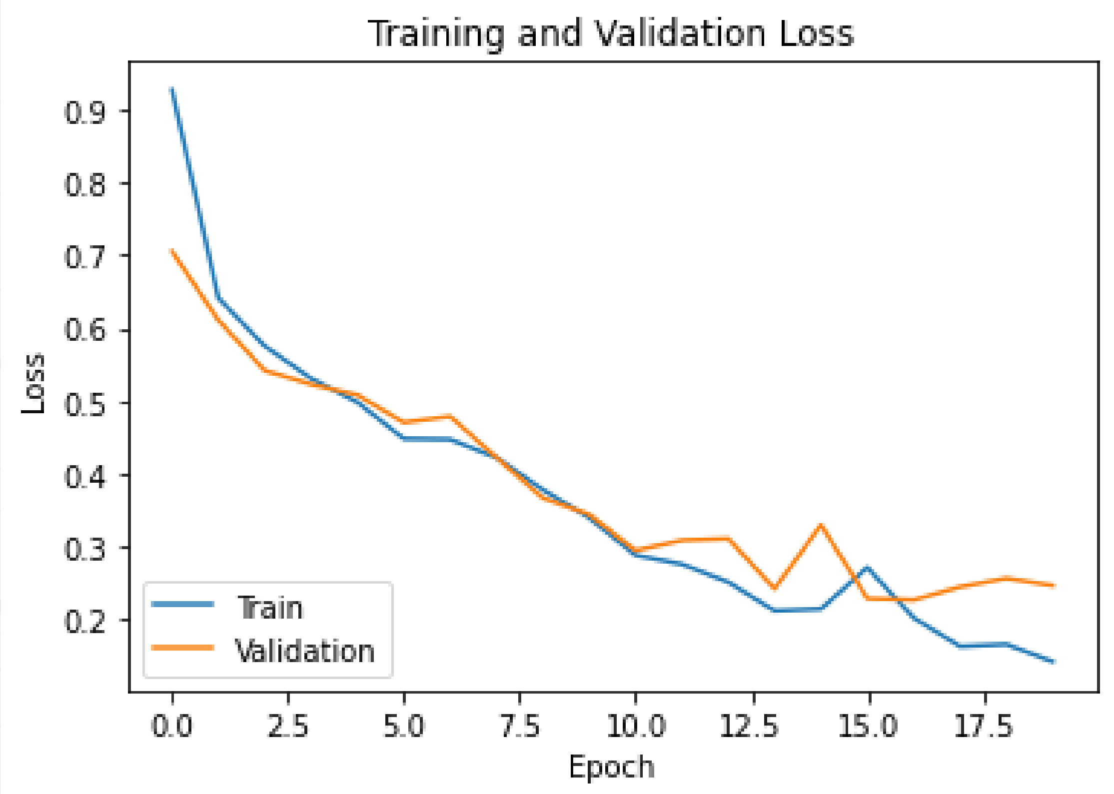
- Final model classifies strokes with high accuracy
- **89.6%** testing accuracy
- **90.3%** validation accuracy
- Demonstrates the use of neural networks as a viable method of stroke classification
- Final model able to return chance of abnormality from cranial CT scan

Extensions

- Testing on larger datasets
- Increasing classes to include other cranial diseases
- Testing during the real-time stroke diagnosis process
- Testing on other types of medical imaging (MRI or microwave imaging)

Appendices


Training and Validation Accuracy
Accuracy (%)
0.95
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch
Train
Validation


Training and Validation Loss
Loss
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Epoch
Train
Validation

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 248, 248, 32)	2624
average_pooling2d (AverageP ooling2D)	(None, 124, 124, 32)	0
conv2d_1 (Conv2D)	(None, 120, 120, 64)	51264
average_pooling2d_1 (Averag ePooling2D)	(None, 40, 40, 64)	0
conv2d_2 (Conv2D)	(None, 38, 38, 128)	73856
average_pooling2d_2 (Averag ePooling2D)	(None, 19, 19, 128)	0
flatten (Flatten)	(None, 46208)	0
dense (Dense)	(None, 64)	2957376
dense_1 (Dense)	(None, 3)	195

Total params: 3,085,315
Trainable params: 3,085,315
Non-trainable params: 0

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 248, 248, 16)	1312
max_pooling2d (MaxPooling2D)	(None, 82, 82, 16)	0
conv2d_1 (Conv2D)	(None, 78, 78, 32)	12832
max_pooling2d_1 (MaxPooling 2D)	(None, 26, 26, 32)	0
conv2d_2 (Conv2D)	(None, 24, 24, 64)	18496
max_pooling2d_2 (MaxPooling 2D)	(None, 8, 8, 64)	0
flatten (Flatten)	(None, 4096)	0
dense (Dense)	(None, 128)	524416
dense_1 (Dense)	(None, 3)	387

Total params: 557,443
Trainable params: 557,443
Non-trainable params: 0

```
accuracy, outputs, spec_acc = testModel(X_test, y_test)
print(accuracy)
print(outputs)
print(spec_acc)
```

```
0.8961038961038961
[1, 0.4173205494880676, 0.028068542480 46875, 0.006431341171264 6484, 2, 2,
[0.9743589743589743, 0.821917808219178, 0.9523809523809523, [15, 1, 0]]
```

$$\boldsymbol{L}(w) = -\sum_{i=1}^{N} y_i \cdot \log(\hat{y}_i)$$

$$\phi(x) = \max(0, x)$$

$$\sigma(\vec{z}) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$