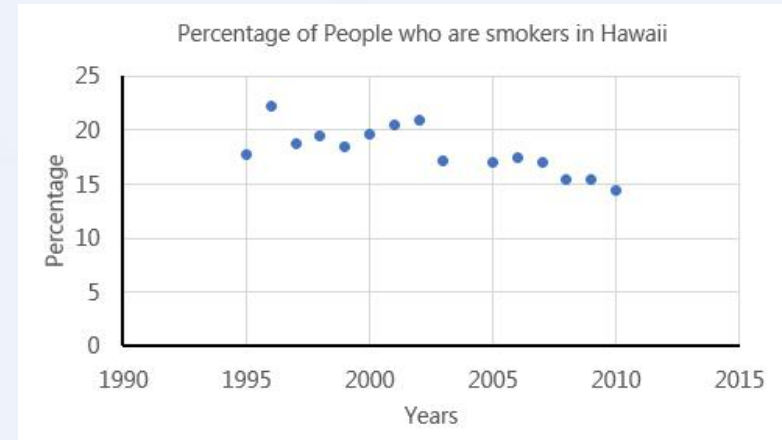
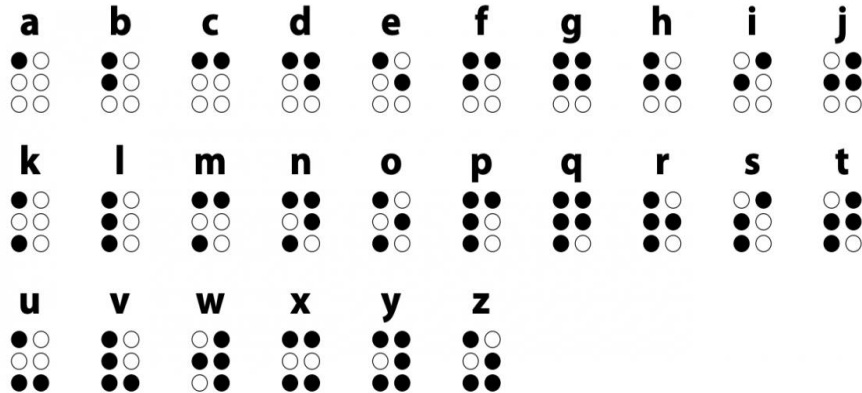


# **A Machine-Learning Based Approach to Chart Image Recognition**

Tan Song Kai

# Background

- By 2050, the Vision Loss Expert Group (VLEG)<sup>1</sup> predicts approximately 61 million people to be blind
- Visual impairment limits access to educational materials
- Braille helps to address the difficulty of reading conventional print
- Resource intensive to transform educational visuals such as graphs



1: [Trends in prevalence of blindness and distance and near vision impairment over 30 years: an analysis for the Global Burden of Disease Study](#)

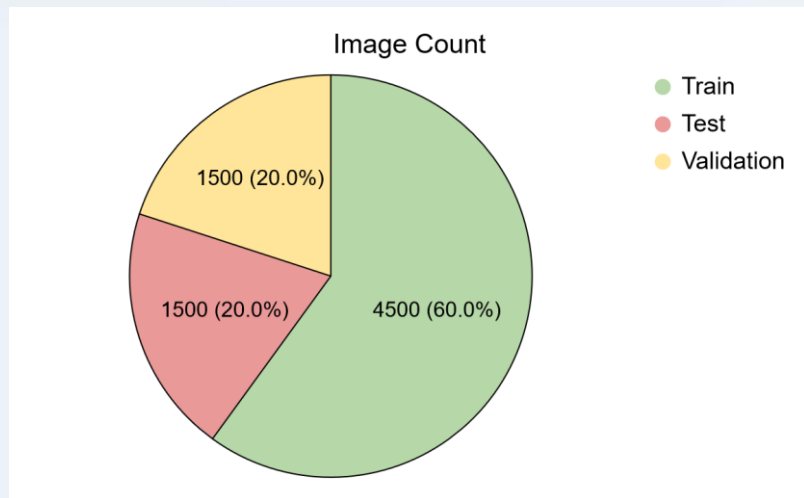
# Problem Statement

- As only a small fraction of educational materials are available for people with visual impairment, we aim to develop a machine learning model to extract chart information to help bridge the gap in learning resources.

# Dataset

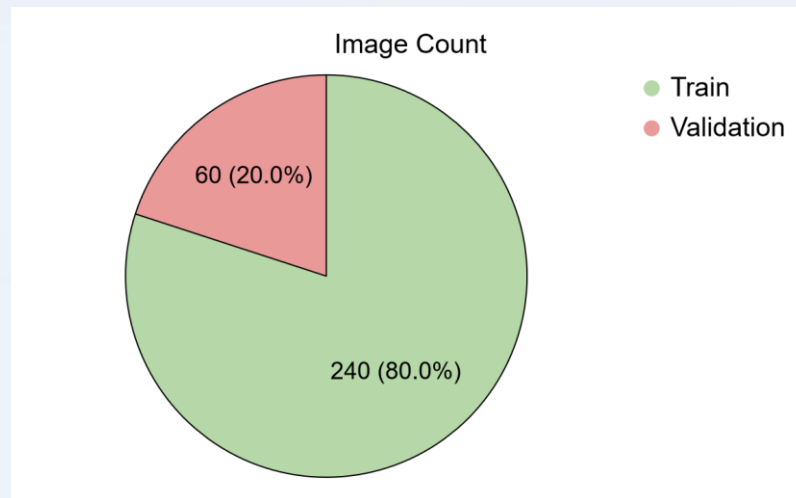
- Kaggle images (<https://www.kaggle.com/competitions/benetech-making-graphs-accessible/data>)

Chart Type



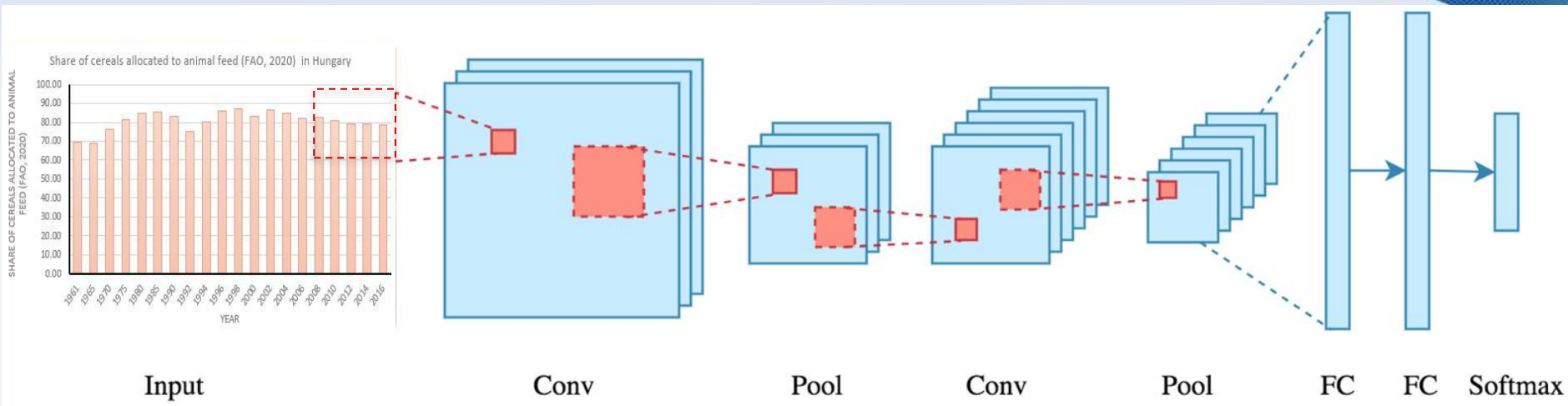
- Chart type: Scatter, Bar, Line
- Ground truth provided and images verified manually

Trend



- Trend: Increasing, Decreasing, Invariant
- Ground truth not provided and images classified manually

# Convolutional Neural Network (CNN)

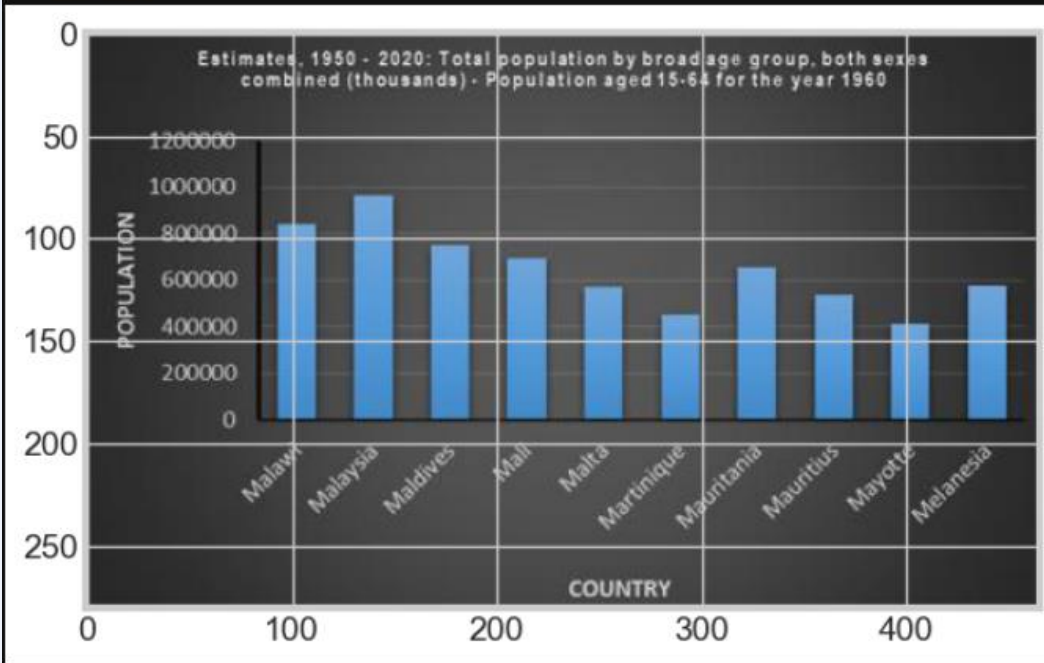


CNN architecture:

- Input
- Convolutional layer
- Pooling layer
- Fully Connected (FC) layer
- Activation Function

# An image can be visualized as a Numpy array

```
plt.imshow(img);
```



```
asarray(img).shape
```

```
(281, 466, 3)
```

```
asarray(img)
```

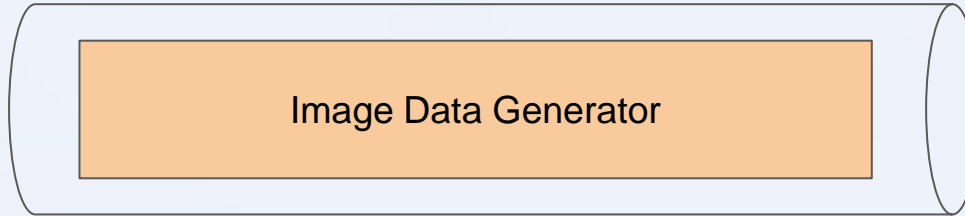
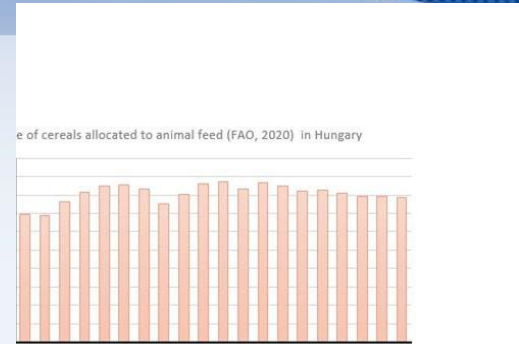
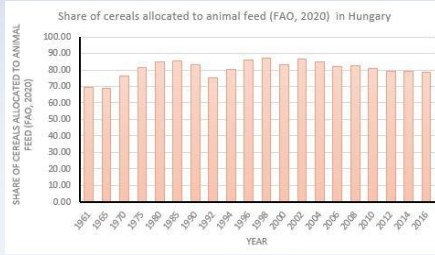
```
array([[39, 39, 39],
       [39, 39, 39],
       [39, 39, 39],
       ...,
       [40, 40, 40],
       [39, 39, 39],
       [39, 39, 39]],
      [[39, 39, 39],
       [39, 39, 39],
       [39, 39, 39],
       ...,
       [40, 40, 40],
       [39, 39, 39],
       [39, 39, 39]],
      [[39, 39, 39],
       [39, 39, 39],
       [39, 39, 39],
       ...,
       [40, 40, 40],
       [39, 39, 39],
       [39, 39, 39]],
      ...,
      [[40, 40, 40],
       [40, 40, 40],
       [40, 40, 40],
       ...,
       [40, 40, 40],
       [38, 38, 38],
       [38, 38, 38]],
      [[40, 40, 40],
       [40, 40, 40],
       [40, 40, 40],
       ...,
       [40, 40, 40],
       [38, 38, 38],
       [38, 38, 38]],
      [[38, 38, 38],
       [38, 38, 38],
       [38, 38, 38],
       ...,
       [40, 40, 40],
       [39, 39, 39],
       [39, 39, 39]]], dtype=uint8)
```

# Modelling Process

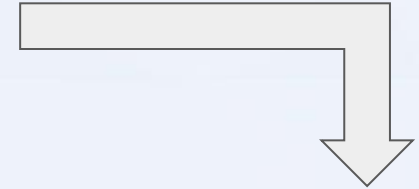
- ImageDataGenerator
  - Batch Generation
  - Boilerplate Code
  - Data Augmentation
- Data Augmentation
  - Transforming original images to produce different copies for training
  - Target class not changed
  - Transformation examples: Translation, Rotation, Zoom, Flipping, Brightness, Cropping

# Modelling Process

Input



Translation  
Rotation  
Zoom  
Flipping  
Brightness  
Cropping  
...



Output

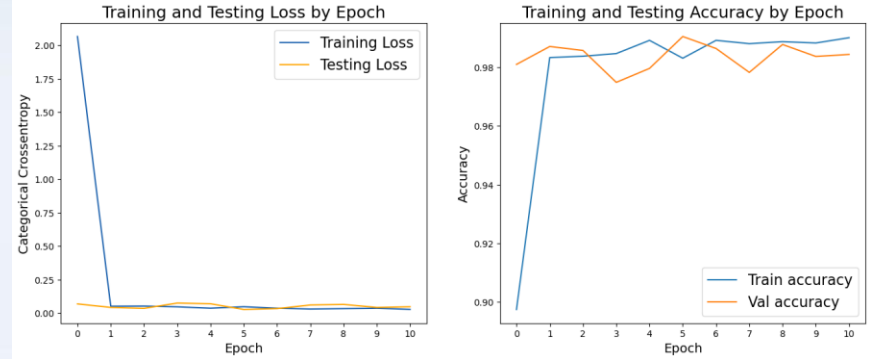
- Chart type
- Trend



# Modelling Results

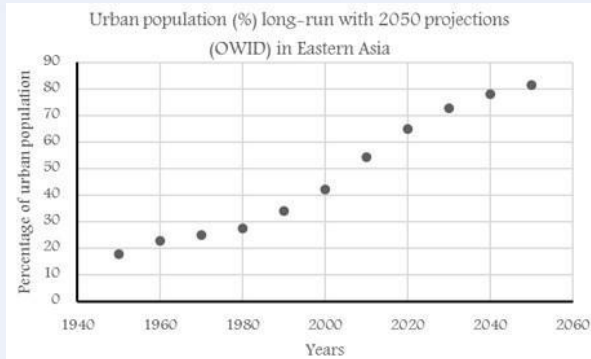
**Baseline: 33.3%**

Model	Train Accuracy	Test Accuracy
AlexNet	82.15%	73.80%
ResNet50	88.71%	95.87%
VGG16	98.31%	99.67%



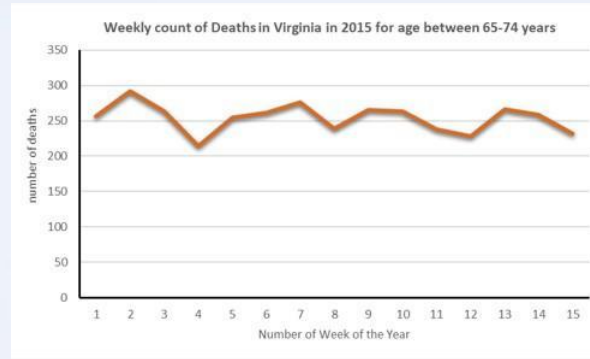
# Modelling Results

## Increasing



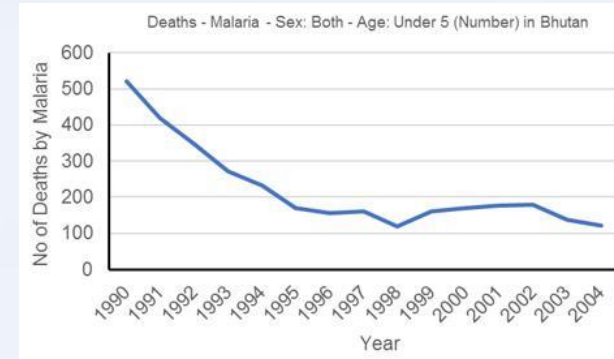
```
{{'decreasing': 0, 'increasing': 1, 'invariant': 2},  
array([[0.0021602, 0.9891059, 0.00873382]], dtype=float32))
```

## Invariant



```
{{'decreasing': 0, 'increasing': 1, 'invariant': 2},  
array([[0.04016925, 0.05962279, 0.900208 ]], dtype=float32))
```

## Decreasing



```
{{'decreasing': 0, 'increasing': 1, 'invariant': 2},  
array([[0.91463304, 0.04081934, 0.04454762]], dtype=float32))
```

# Limitations

- Manually labelling chart trends can differ between individuals
- Chart trends may not be entirely conclusive
- Hard to extract text using Pytesseract or OpenCV

# Conclusion & Recommendations

- Feasible to classify chart types using CNN
- Transfer learning from VGG16/ResNet50 is more effective than training network from scratch

## Moving forward

- Classification of different chart images (pie charts, area charts, Venn diagrams etc)
- To extract more features from plots (axis names/values, legend etc)
- Chart value interpolation → curve fitting
- OCR to create captions for charts

**Thank you for your attention!**  
**Q&A**