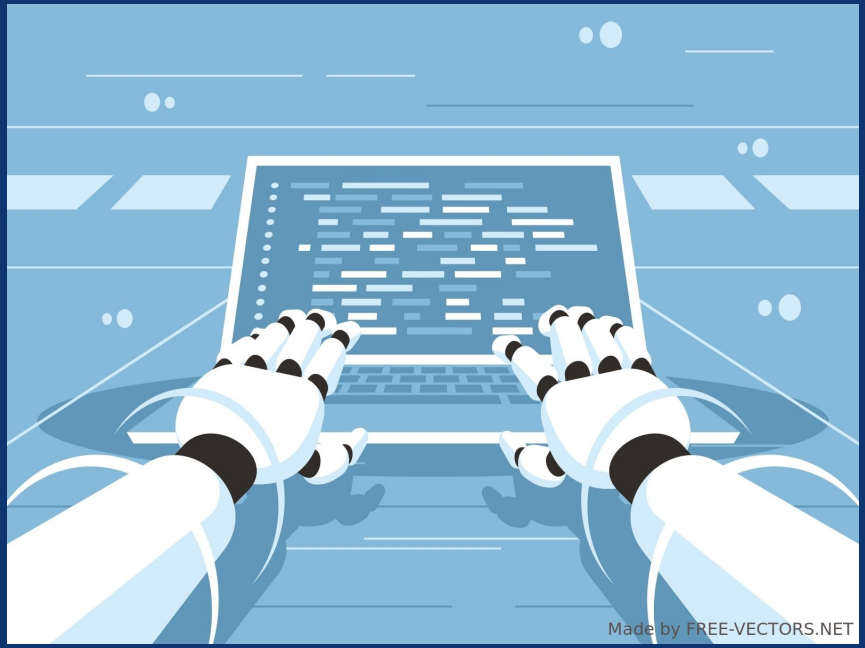
GENERATING AND IMPLEMENTING CNN MODELS TO DETECT ASL NUMBERS

Justin Caringal
Truman State University
Kirksville, Missouri, USA

INTRODUCTION

- Artificial intelligence
- Leverage resources

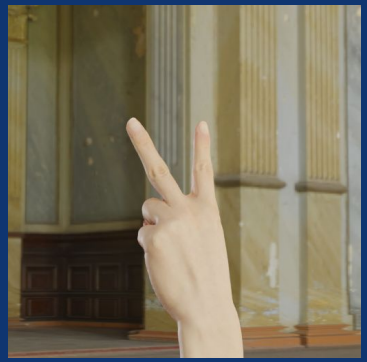


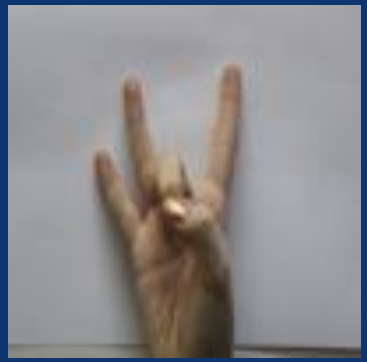
PROBLEM

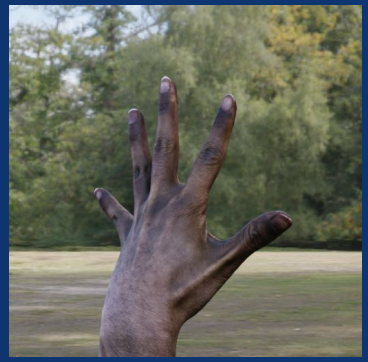
Detect and classify ASL numbers 0-9

DATASET

Detect and classify ASL numbers 0-9





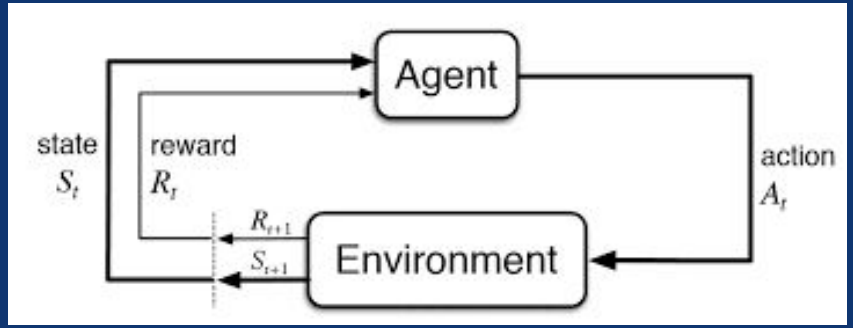


kaggle 11,762 images

BASICS OF DEEP CONVOLUTIONAL Q-LEARNING

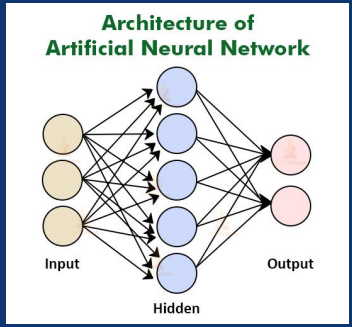
Q-LEARNING

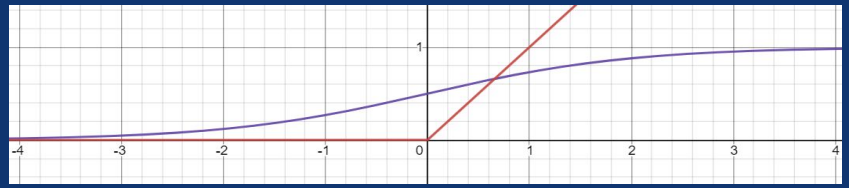
- State (Q-value)
- Reward (Temporal Difference)
- Action (Bellman's equation)

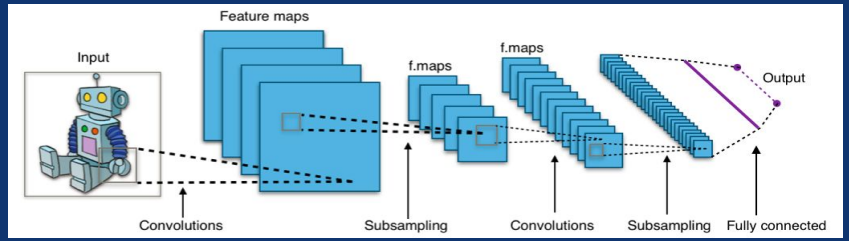


DEEP CONVOLUTION

- Deep
  - Neural Network
  - Memory
- Layers
  - Convolution
  - Max Pooling



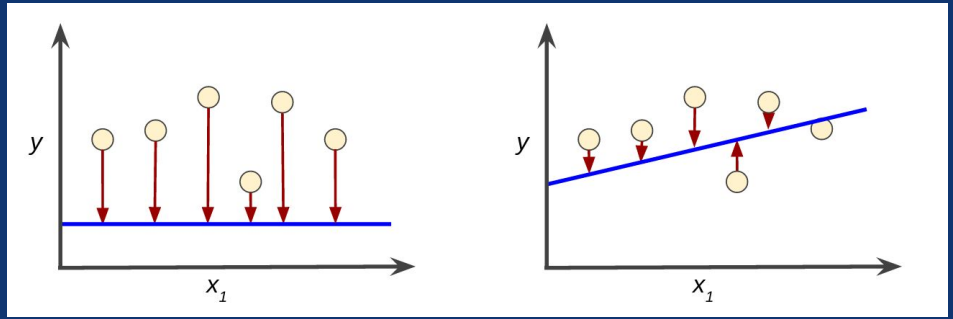




SOLUTION

MODEL METRICS

- Validation Accuracy
- Validation Loss
- Testing Accuracy
- Testing Loss



EARLY MODEL GENERATION

- 2,062 images (Mavi dataset) kaggle
  - 1,962 training, 100 validation
- 512×512 pixels
- Various layers & parameters tested K
  - 32×64×32 Conv2D (3×3)
  - 2×2 MaxPooling2D
  - Dense
  - Dropout
- Early Stopping callback

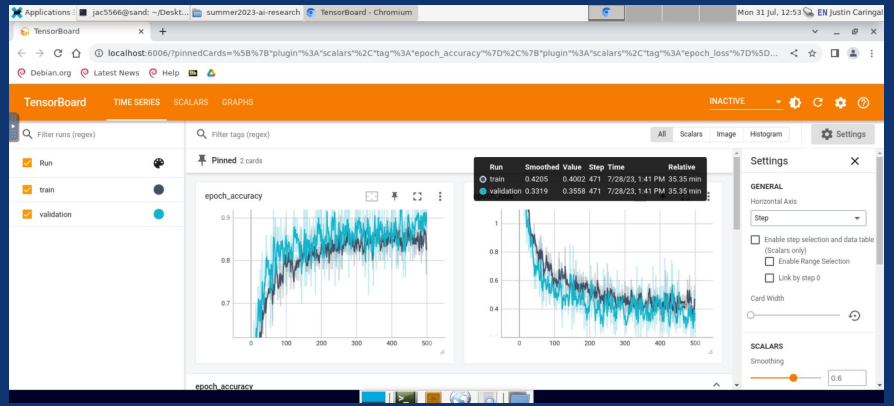
EARLY STOPPING

- Measures improvements
- Stops if no improvement within *n* epochs

```
# Added Early Stopping
my_callback = [EarlyStopping(
    monitor = MONITOR,
    min_delta = MIN_DELTA,
    patience = PATIENCE,
    mode = 'auto',
    baseline = 1,
    restore_best_weights = True)]
```

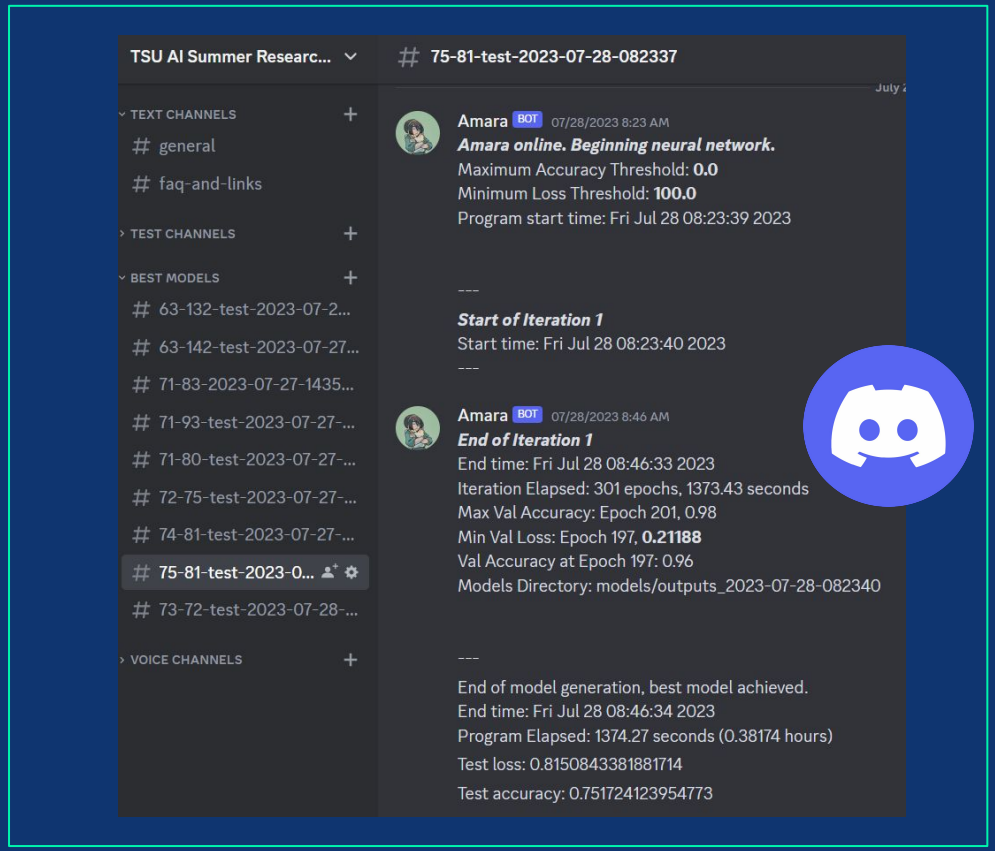
VARIOUS IMPROVEMENTS

- 11,762 images (Mavi, Thakur, Lexset), 224×224 kaggle
  - 8,712 training, 2,905 validation, and 145 testing
- Elapsed time
- Discord bot integration
- Introduction of testing statistics
- Callbacks
  - CSVLogger
  - LearningRateScheduler
  - TensorBoard



DISCORD.PY

- Data collection
- Better organization
- Information dispersal
- Automatic channel generation
  - One-click execution

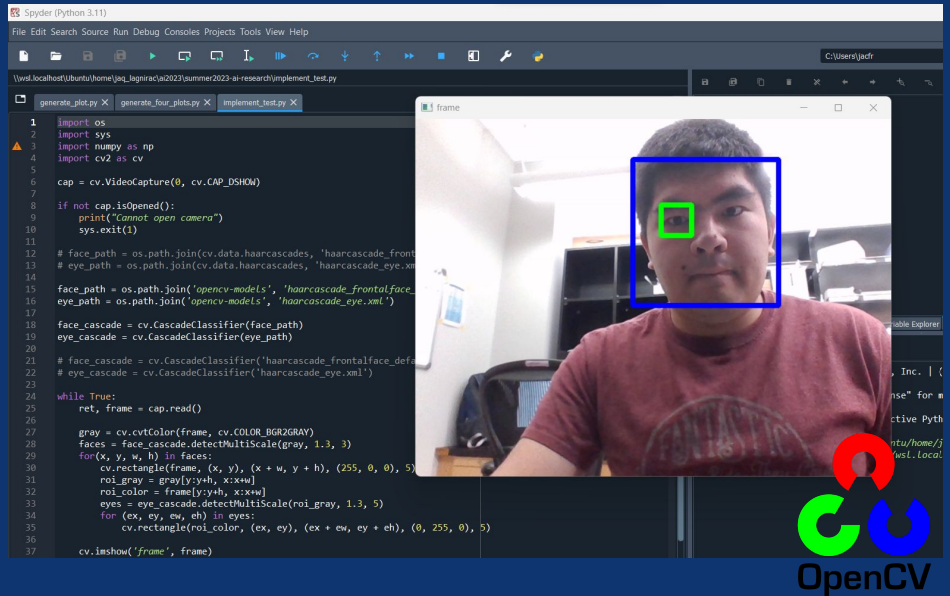


TENSORFLOW MODEL ZOO

- Pre-trained models
- MobileNetv2
  - Trained on ImageNet
  - Freezing layers
    - Had little effect
  - Connected by GlobalAveragePooling2D
  - +10% accuracy, halved loss
- Other models not tested as of writing
  - Ex. Resnet50

OPENCV

- Computer vision framework
- Haar Cascade Classifier
- Need to connect generated model



DATA

DATA COLLECTED

Model	*Code*	*Val. Acc. (%)*	*Val. Loss (%)*	*Test Acc. (%)*	*Test Loss (%)*	*Elapsed (hrs.)*
C/BFS	20-131023	94	16.1	60	140.3	2.67
C/BFS	20-160248	94	16.2	48.3	161.3	1.77
C/BFS	20-182231	90	20.7	63.4	132.7	1.53
C/BFS	25-113027	100	3.9	62.1	128.9	1.55
C/BFS	26-192442	98	13.9	46.9	203.1	1.35
C/BFS	27-101838	96	8.8	63.4	142.5	1.19
MNv2	27-143535	94	32	71.7	83.5	0.25
MNv2	27-154003	94	32.9	71.7	93	0.17
MNv2	27-162115	98	19	71	80	0.25
MNv2	27-164541	94	20.1	72.4	75.6	0.32
MNv2	27-215322	100	1.4	74.5	81.2	0.36
MNv2	28-082337	96	21.2	75.2	81.5	0.38
MNv2	28-110102	96	20.7	73.1	72.6	0.35

C/BFS - Custom/Built-From-Scratch
MNv2 - MobileNetv2
Percents rounded to nearest tenth, hours rounded to nearest hundredth

CONCLUSION

- Problem researched
- Solution developed
- Added to the workflow pipeline
- Room for improvements

FUTURE DIRECTIONS

- Full OpenCV Implementation
- Script/Tool Suite
- Configuration Files

ACKNOWLEDGEMENTS

- Dr. Ruthie Halma
- Dr. Don Bindner
- Sam Myers

FOR MORE INFORMATION:

DISCORD

Access Discord server
with outputs

GITHUB

Visit Github repository
for full commit tree


slidesgo