Greek Sign Language Letter Recognition

DEMOKRITOS
NATIONAL CENTRE FOR SCIENTIFIC RESEARCH

ΠΑΝΕΠΙΣΤΗΜΙΟ
ΠΕΛΟΠΟΝΝΗΣΟΥ
UNIVERSITY of the PELOPONNESE

Marogianni Sofia 2022202204010
Papadopoulos Nikolas 2022202204025

https://github.com/cucuvaya/gsl-fingerspelling

Presentation Outline

I Introduction to the Problem

II Dataset Creation

III CNN Model

IV VGG Transfer Model

V Final Evaluation

VI Demonstration

Introduction to the Problem

Introduction to the Problem

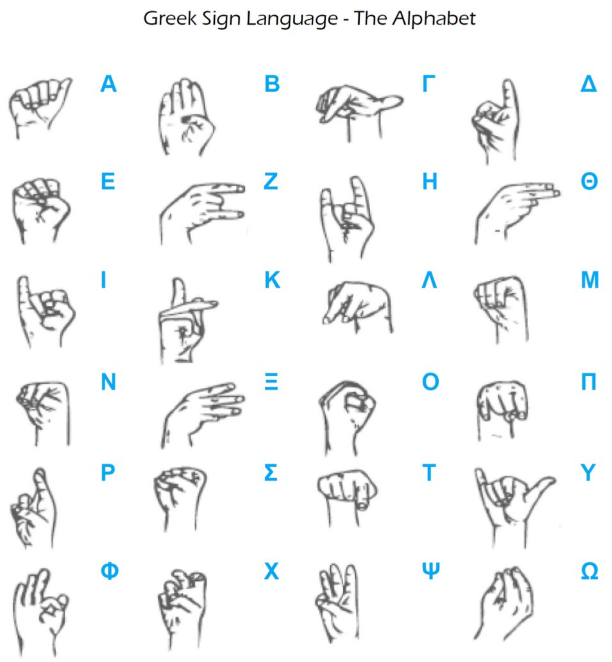
Greek Sign Language

- Legally recognized as the official language area of the Deaf community for educational purposes in Greece since 2000.
- Estimated to be used by some 40,600 people.
- Each country develops its own sign language with fundamentally different meanings and a different alphabet.
- The educational material that exists in the Greek Sign Language is limited.

Introduction to the Problem

The Project

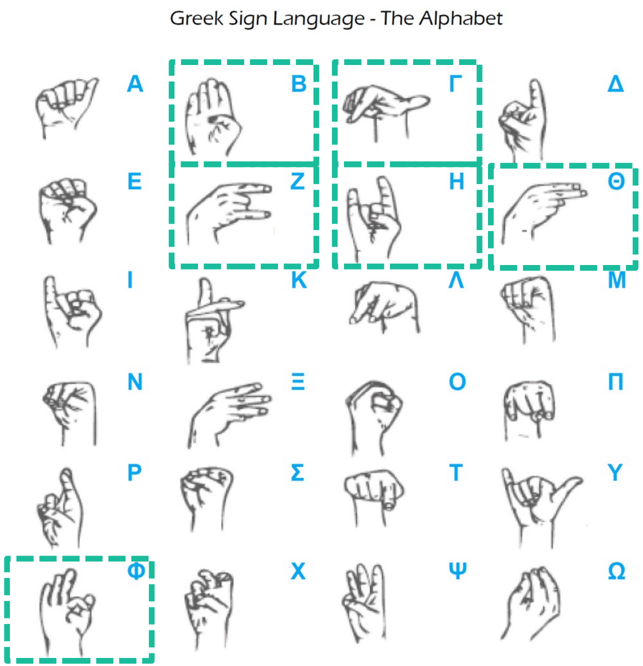
- Recognizing Greek sign letters live.
- Only prerequisite is a camera.
- We choose 6 letters (Β, Γ, Θ, Η, Ζ, Φ) to train and test our model.



Introduction to the Problem

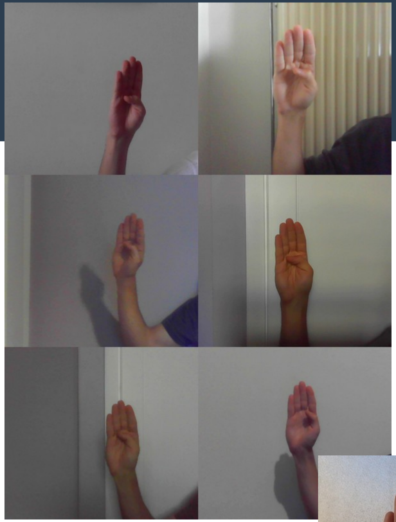
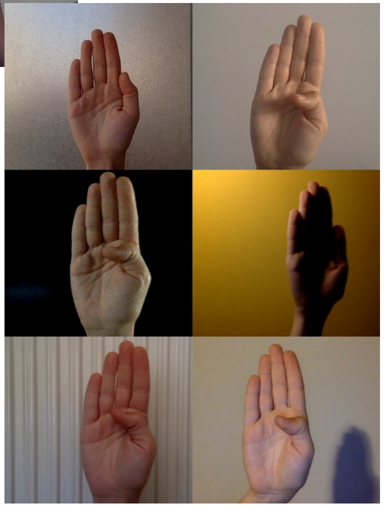
The Project

- Recognizing Greek sign letters live.
- Only prerequisite is a camera.
- We choose 6 letters (Β, Γ, Θ, Η, Ζ, Φ) to train and test our model.



Dataset Creation

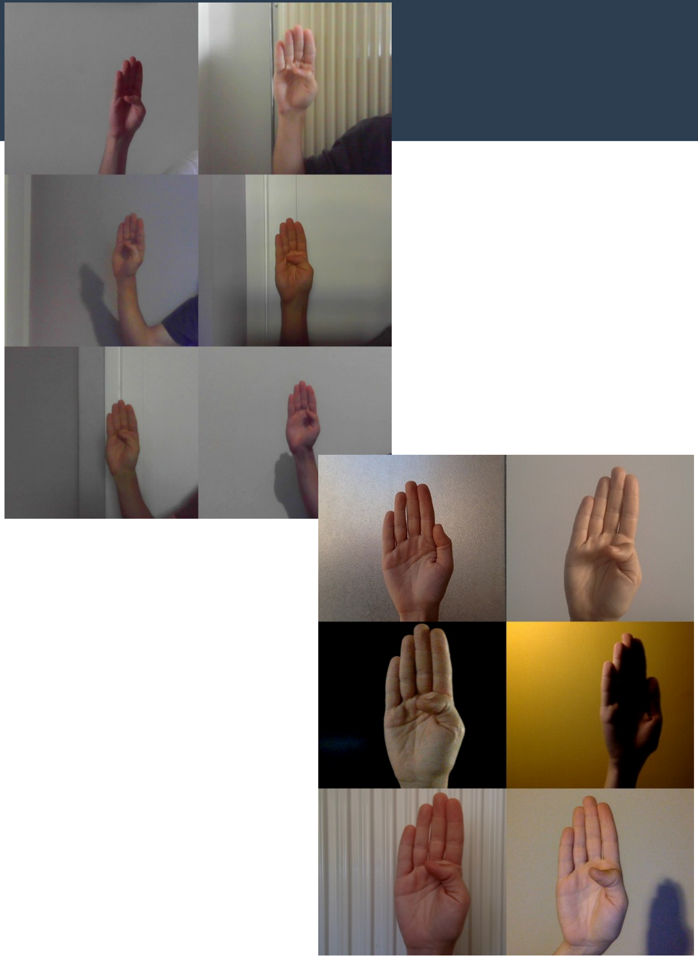
Dataset Creation

- **Unable to find suitable GSL dataset.**
- **Decided to create it:**
  - → **Two signers**
  - → **Two days**
  - → **Three sessions per day (morning, evening, night)**
  - → **50/100 photos per letter in each session**
  - → **Difference in:**
    - **Background (colors, textures)**
    - **Lighting (strong/dim, natural/technical)**
    - **Distance from camera**
    - **Signing style/variation**

Dataset Creation

- **Collected dataset consists of:**
  - 5400 rgb pictures (1800 + 3600)
- **Two forms of final dataset:**
  - 5400 rgb pictures (224*224 pixels) → VGG Input
  - 5400 grayscale pictures (28*28 pixels) converted to '.csv' format → CNN Input
- **Final dataset was split to:**
  - 6 training sessions
  - 4 validation sessions
  - 2 test sessions



CNN Model

CNN Model

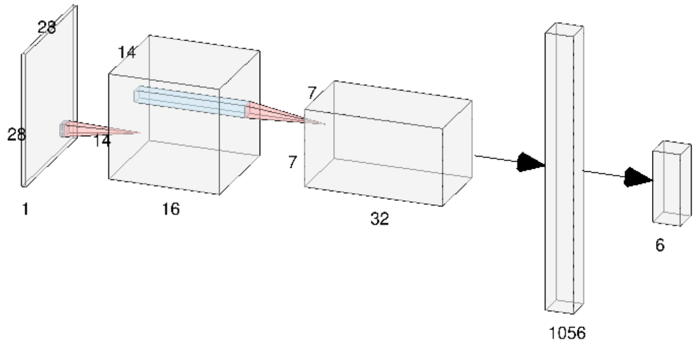
Optimization plan:

- Small depth → Large depth
- Different batch sizes (32 → 2700)
- Tune learning rate (0.01-0.0001)
- Regularization if needed (Dropout-L2)

CNN Model

Baseline architecture:

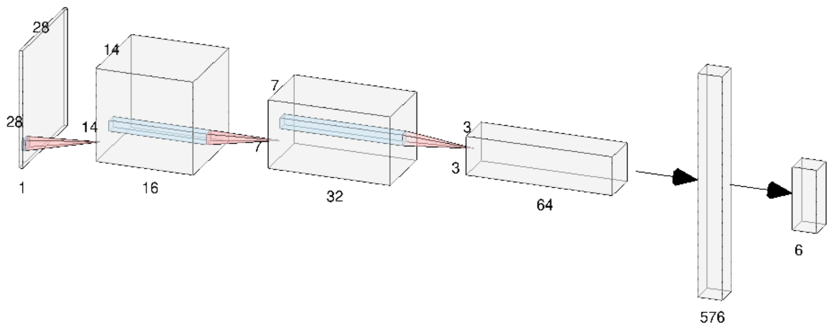
- Conv(1 → 16 → 32) → Linear(1056 → 6)
- Each Convolutional layer consists of:
  - Conv2d(d_in, d_out, kernel=3, stride=1, padding=1)
  - BatchNorm2d(d_out)
  - ReLU(inplace=True)
  - Dropout(d)
  - MaxPool2d(kernel=2, stride=2)



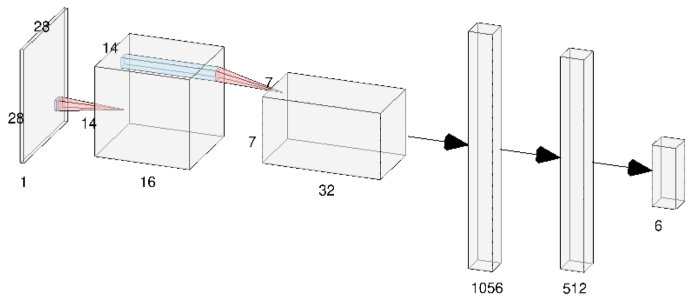
CNN Model

Other architectures:

- Conv(1 → 16 → 32 → 64) → Linear(1056 → 6)



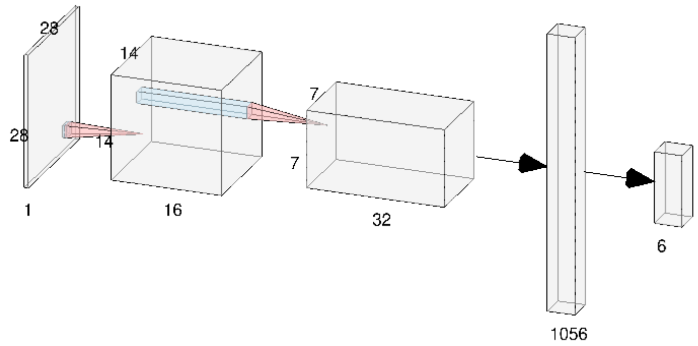
- Conv(1 → 16 → 32) → Linear(1056 → 512 → 6)



CNN Model

Best Model:

- Baseline architecture
- Lr = 0.001
- batch_size = 1024
- n_epochs = 100
- Patience = 15
- Dropout = 0.
- No L2-Regularization



CNN Model

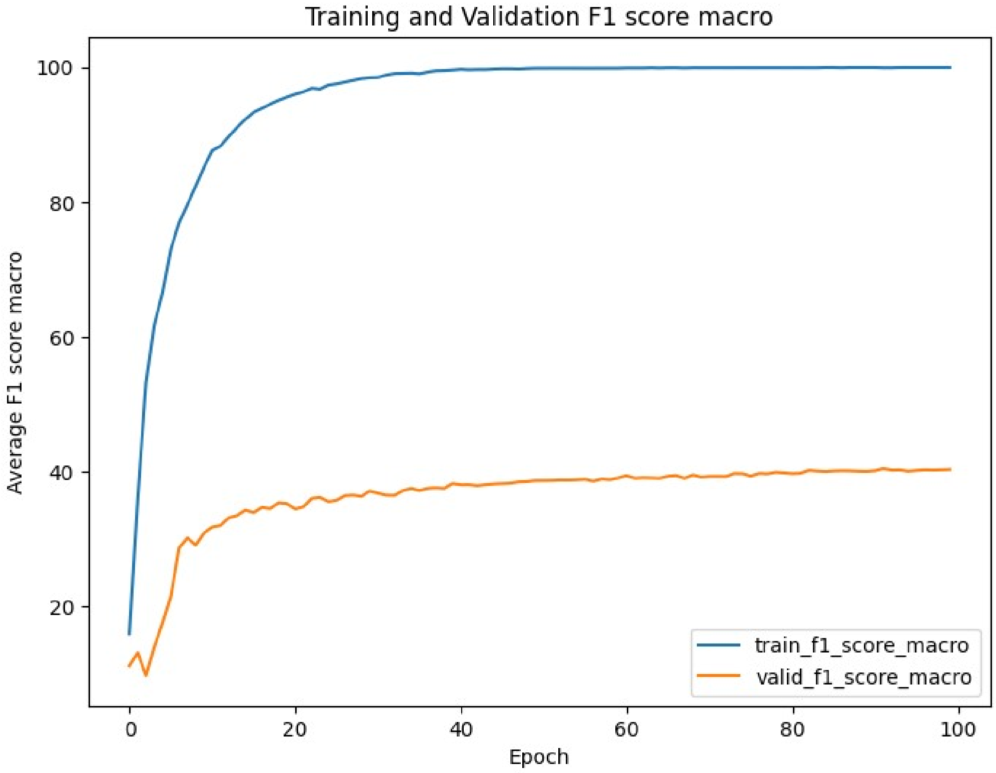
Best Model:

- Baseline architecture
- Lr = 0.001
- batch_size = 1024
- n_epochs = 100
- Patience = 15
- Dropout = 0.
- No L2-Regularization

CNN Model

Best Model:

- Baseline architecture
- Lr = 0.001
- batch_size = 1024
- n_epochs = 100
- Patience = 15
- Dropout = 0.
- No L2-Regularization



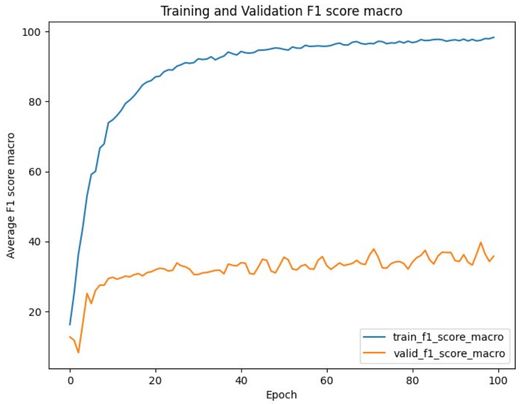
CNN Model

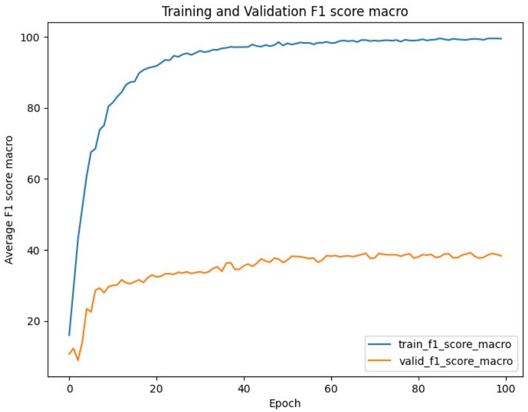
Best Model:

- Baseline architecture
- Lr = 0.001
- batch_size = 1024
- n_epochs = 100
- Patience = 15
- Dropout = 0.
- No L2-Regularization

CNN Model

- Efforts to increase generalization and decrease overfitting in the training data:
  - Dropout: (0.2 – 0.6)
  - L2 regularization





- Different architectures did not produce better results!

CNN Model

Best CNN Model:

- "valid_f1_score_macro_best": 0.4036
- "valid_best_acc": 0.3944

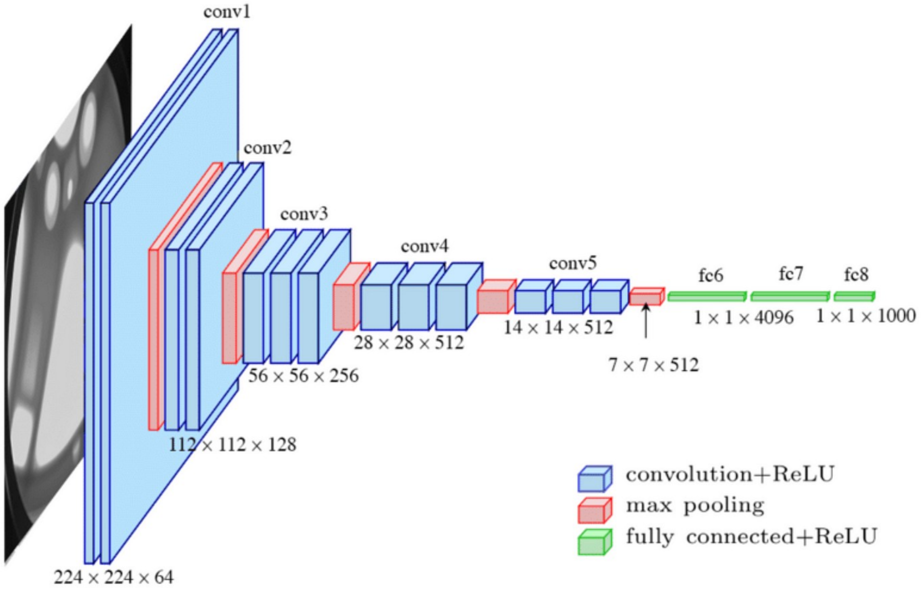
Letter	F1_score
Β	0.44
Γ	0.31
Ζ	0.36
Η	0.52
Θ	0.35
Φ	0.42

VGG Transfer Model

VGG Transfer Model

Transfer Learning Method

- Take the VGG model pretrained
- Freeze all the layers
- Replace the final decision layer (fc8) with one corresponding to 6 class output
- Hyperparameter Optimization
- Gradual unfreezing of pretrained classification layers to achieve the best model



VGG Transfer Model

Optimization Plan

- 1 Unfrozen Classification Layer → 3 Unfrozen Classification Layers
- Different batch sizes (32-256)
- Tune learning rate (0.01-0.0001)
- Dropout in the last custom layer (0.2-0.4)

VGG Transfer Model

Two candidate models

VGG-Transfer_full_unfrozen:

- 3 of the 3 classification layers trainable
- Lr = 0.001
- batch_size = 64
- n_epochs = 100
- Patience = 20
- Dropout = 0.2

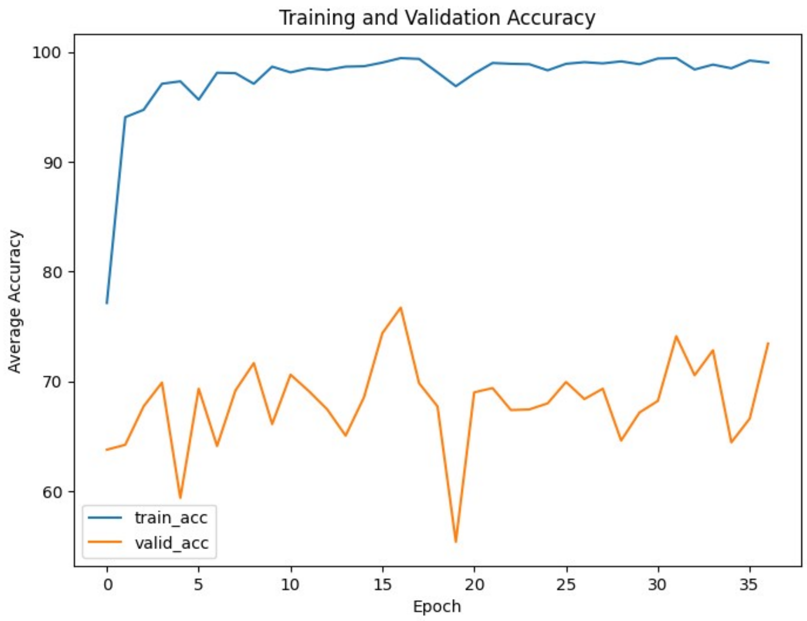
VGG-transfer_2_unfrozen:

- 2 of the 3 classification layers trainable
- Lr = 0.001
- batch_size = 248
- n_epochs = 100
- Patience = 10
- Dropout = 0.4

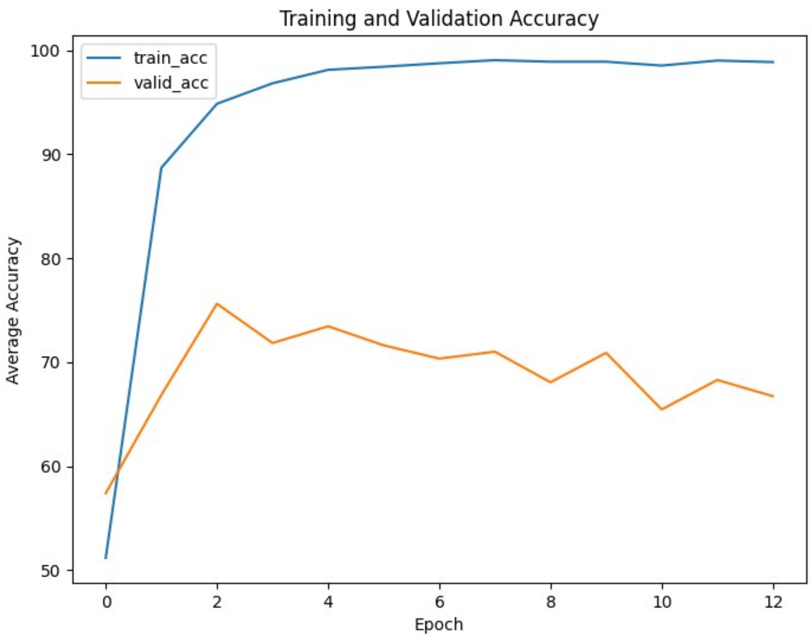
VGG Transfer Model

Two candidate models

VGG-Transfer_full_unfrozen:



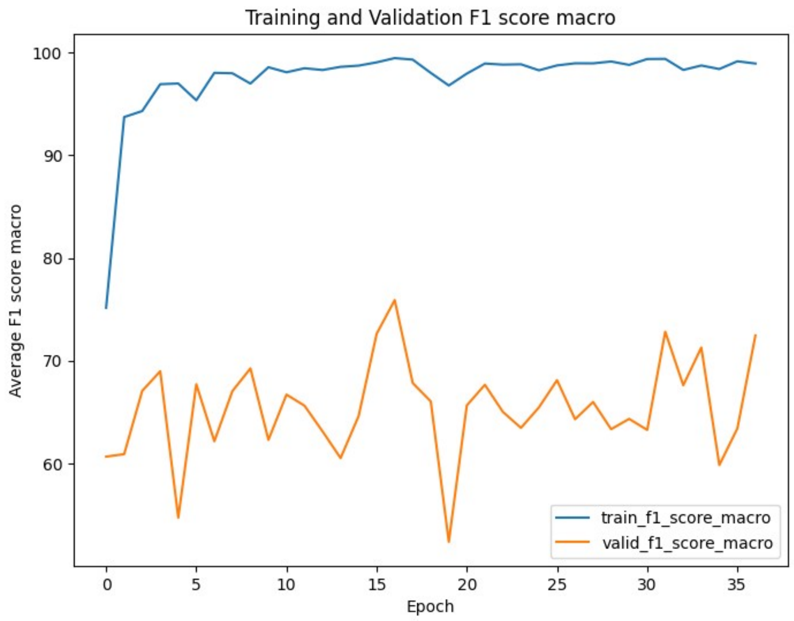
VGG-transfer_2_unfrozen:



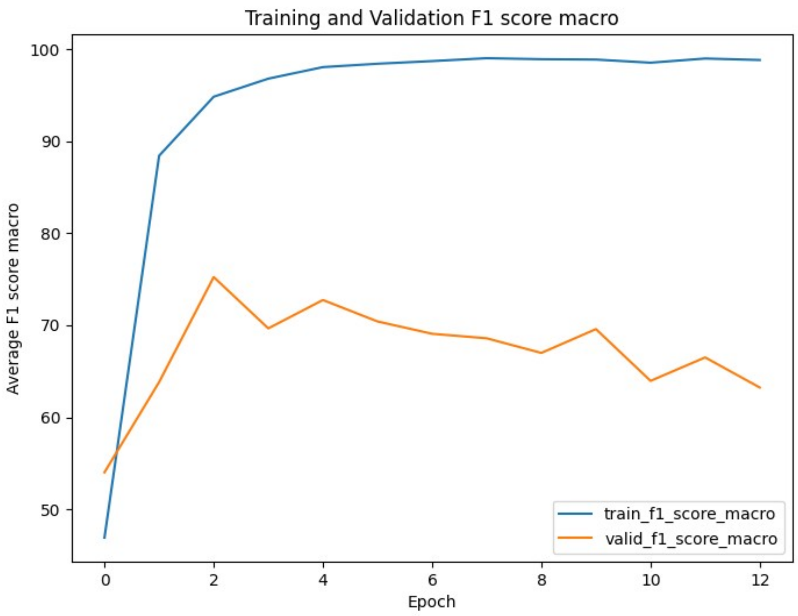
VGG Transfer Model

Two candidate models

VGG-Transfer_full_unfrozen:



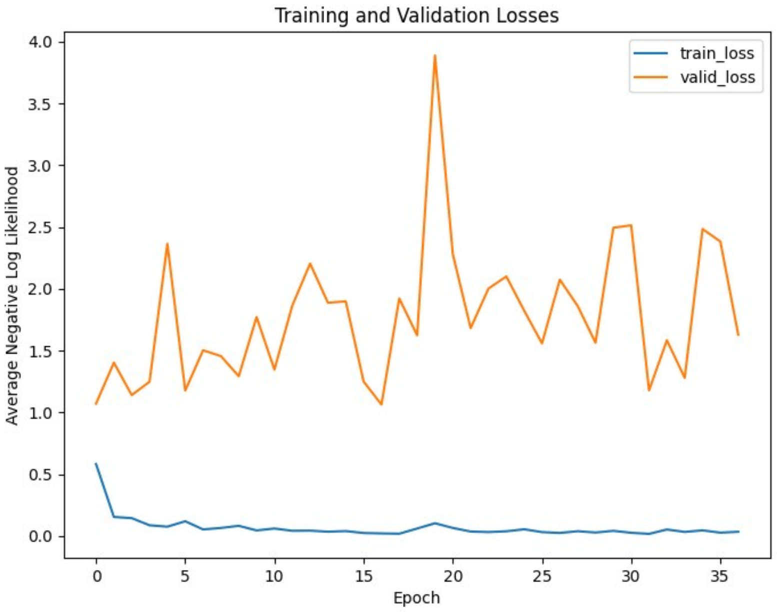
VGG-transfer_2_unfrozen:



VGG Transfer Model

Two candidate models

VGG-Transfer_full_unfrozen:



VGG-transfer_2_unfrozen:

VGG Transfer Model

Two candidate models

VGG-Transfer_full_unfrozen:

- "valid_f1_macro_best": 0.76
- "valid_best_acc": 0.77

Letter	F1_score
Β	0.69
Γ	0.85
Ζ	0.65
Η	0.92
Θ	0.69
Φ	0.76

VGG-transfer_2_unfrozen:

- "valid_f1_macro_best": 0.75
- "valid_best_acc": 0.76

Letter	F1_score
Β	0.6
Γ	0.89
Ζ	0.72
Η	0.94
Θ	0.68
Φ	0.67

VGG Transfer Model

Better All Around!

Two candidate models

VGG-Transfer_full_unfrozen:

- "valid_f1_macro_best": 0.76
- "valid_best_acc": 0.77

Letter	F1_score
B	0.69
Γ	0.85
Z	0.65
H	0.92
Θ	0.69
Φ	0.76

VGG-transfer_2_unfrozen:

- "valid_f1_macro_best": 0.75
- "valid_best_acc": 0.76

Letter	F1_score
B	0.6
Γ	0.89
Z	0.72
H	0.94
Θ	0.68
Φ	0.67

Final Evaluation

Final Evaluation - CNN Best Model

Candidate Models on Test Set:

- **CNN Model**
- "test_f1_score_macro_best": 0.56
- "test_best_acc": 0.55

Letter	F1_score
Β	0.54
Γ	0.3
Ζ	0.52
Η	0.66
Θ	0.77
Φ	0.55

Final Evaluation – VGG Transfer Models

Candidate Models on Test set:

VGG-Transfer_full_unfrozen:

- "test_f1_macro_best": 0.85
- "test_best_acc": 0.86

Letter	F1_score
Β	0.84
Γ	0.97
Ζ	0.84
Η	0.93
Θ	0.74
Φ	0.76

VGG-transfer_2_unfrozen:

- "test_f1_macro_best": 0.74
- "test_best_acc": 0.77

Letter	F1_score
Β	0.82
Γ	0.97
Ζ	0.72
Η	0.89
Θ	0.41
Φ	0.6

Final Evaluation – VGG Transfer Models

Our Final Choice!

Candidate Models on Test set:

VGG-Transfer_full_unfrozen:

- "test_f1_macro_best": 0.85
- "test_best_acc": 0.86

Letter	F1_score
Β	0.84
Γ	0.97
Ζ	0.84
Η	0.93
Θ	0.74
Φ	0.76

VGG-transfer_2_unfrozen:

- "test_f1_macro_best": 0.74
- "test_best_acc": 0.77

Letter	F1_score
Β	0.82
Γ	0.97
Ζ	0.72
Η	0.89
Θ	0.41
Φ	0.6

Demonstration

Demonstration

Demo: https://github.com/cucuvaya/gsl-fingerspelling/blob/main

Future Improvements

- Gather a larger dataset:
  - More captures
  - Add more signers
  - Include more letters
  - More diverse backgrounds
- Integrate hand tracking with ‘Mediapipes’ to improve results
- Cross validation through different train/validation/test sets
- Integrate a voting technique on N number of frames

References

- https://github.com/Leo-xxx/pytorch-notebooks/blob/master/Torn-shirt-classifier/VGG16-transfer-learning.ipynb
- https://github.com/tyiannak/dl-python/blob/main/cnn_1.py
- https://github.com/Mirwe/Real-time-ASL-alphabet-recognition/blob/main/hand_tracking.py