

Hochschule  
München  
University of  
Applied Sciences

Fakultät 7

Anna Reiter, Chiara Perocco

# Advanced Deep Learning

Team Project



# OVERVIEW

## 1. Introduction

- 1.1 Architecture of the system
- 1.2 Object Detector
- 1.3 Image Classifier
- 1.4 Article Agent
- 1.5 Article Assembler
- 1.6 Diffusion Model

## 2. Results

## 3. Conclusion

## 4. Outlook

# Domain

# Domain

Domain: informative text about American Sign Language (ASL)



Article Structure:

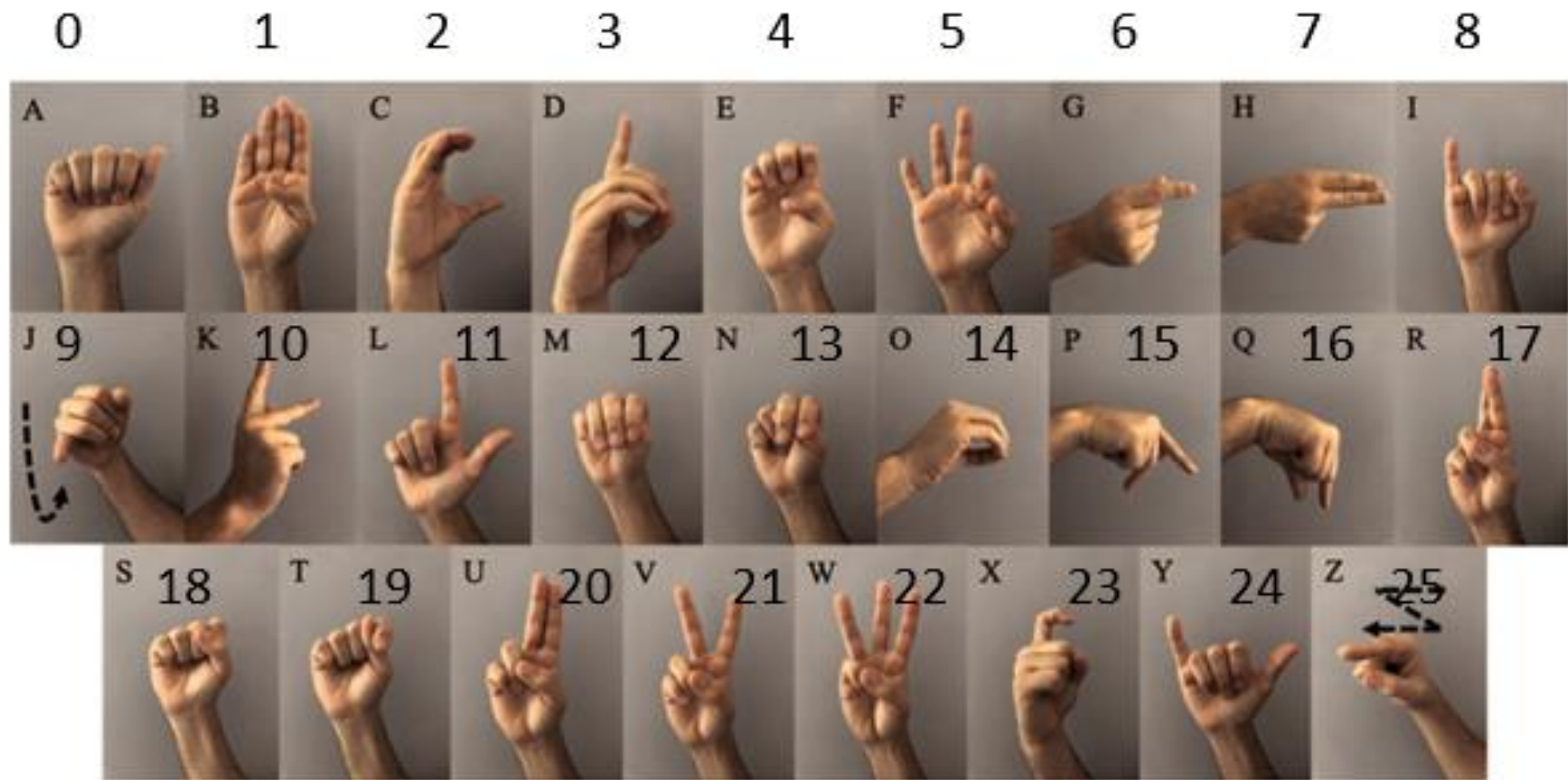
1. Introduction
2. The letter in written language
3. The letter in sign language
4. Conclusion

} Each paragraph is followed by an image

Dataset: ASL(American Sign Language) Alphabet Dataset → consists of 29 classes (only 26 used)

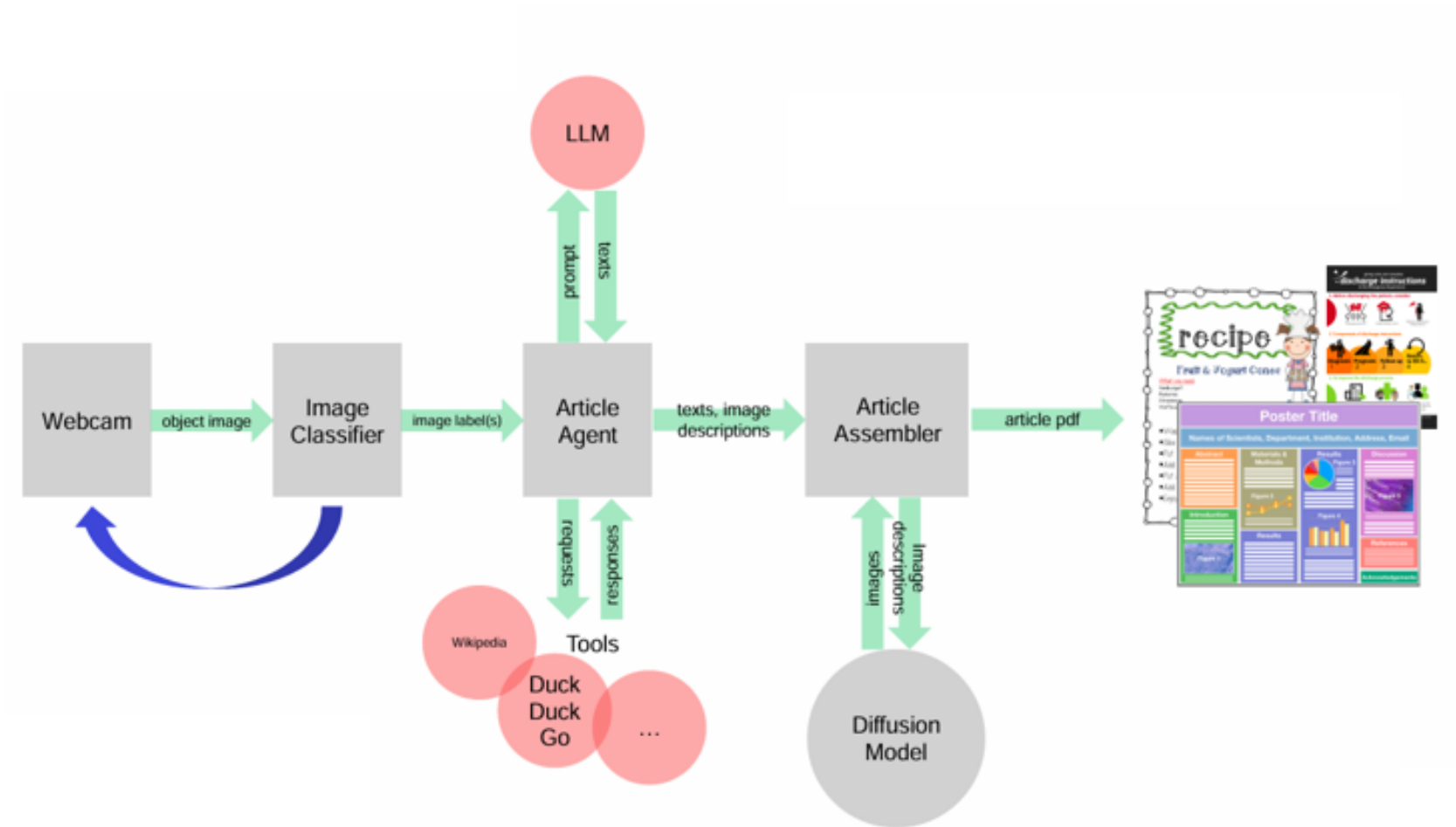
Source: Kaggle ([ASL\(American Sign Language\) Alphabet Dataset](#))

# Domain



# Architecture of the system

# Architecture of the system



# Object Detector



# Object Detector

- Laptop camera
- Opencv-python
- Takes a photo by triggering the „y“ (yes) on keyboard
- Takes no photo by triggering the „n“ (no) on keyboard

# Image Classifier

# Image Classifier

- AlexNet from scratch
- VisionTransformer pretrained on ImageNet
- ResNet50 pretrained on ImageNet

# Article Agent, Article Assembler, Diffusion Model

# Article Agent, Article Assembler, Diffusion Model

- Tools: Wikipedia, DuckDuckGo
- LLM: Llama3.1 von Ollama
- Markdown and Pandoc
- HuggingFace Model: kakaobrain/karlo-v1-alpha

# Results

## 1. Ds1

1.1 AlexNet

1.2 ResNet50

1.3 VisionTransformer

## 2. Ds2

2.1 AlexNet

2.2 ResNet50

2.3 VisionTransformer

# Definition Ds1, Ds2

## Ds1

300 images per class (80% train, 10% val, 10% test)  
→ computing power

Data augmentation:

- horizontal flip
- crop (padding  $n = 4$ )

Hyperparameter tuning with optuna

## Ds2

Adjustment of ds1 with mediapipe; 1300 images per class (80% train, 10% val, 10% test)

1. Data augmentation:

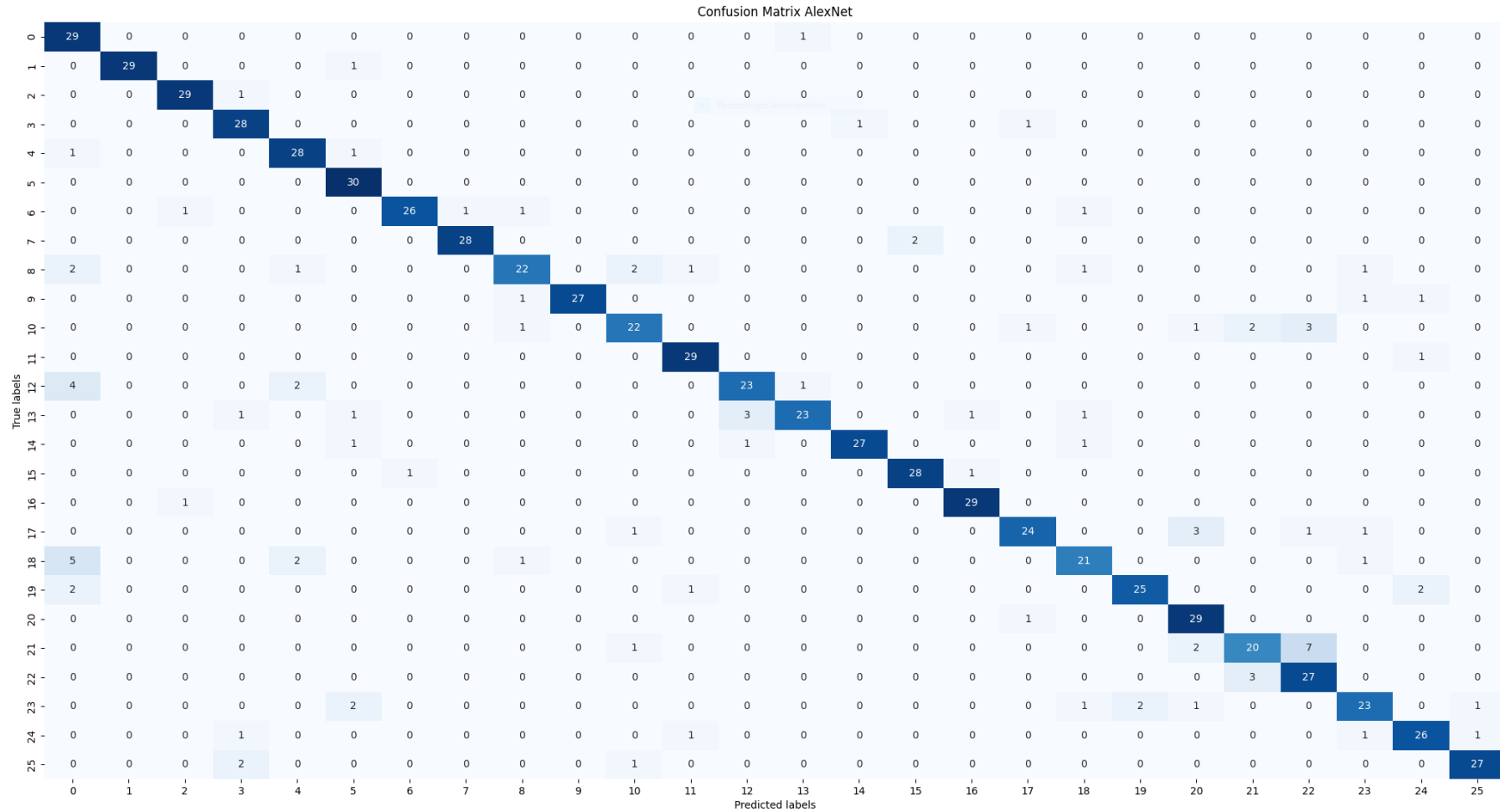
- horizontal flip
- crop (padding  $n = 4$ )

2. Data augmentation:

- Rotation
- Colour properties
- Affine transformations
- Grayscale conversion

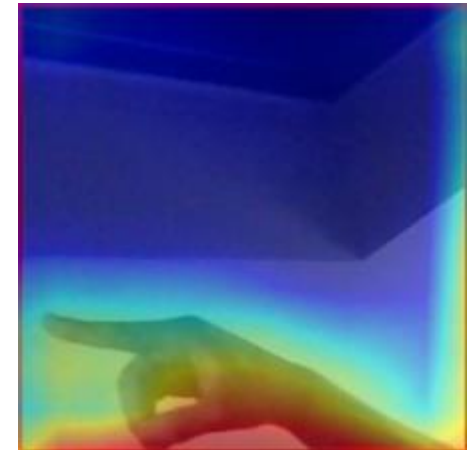
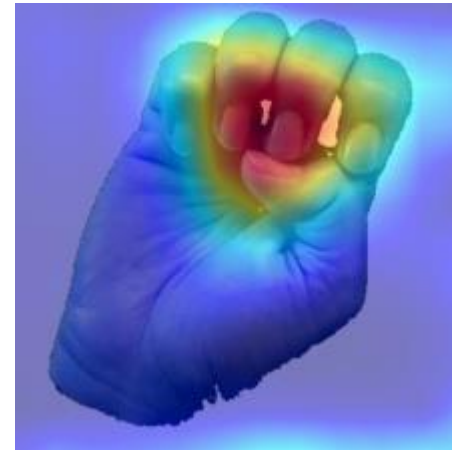
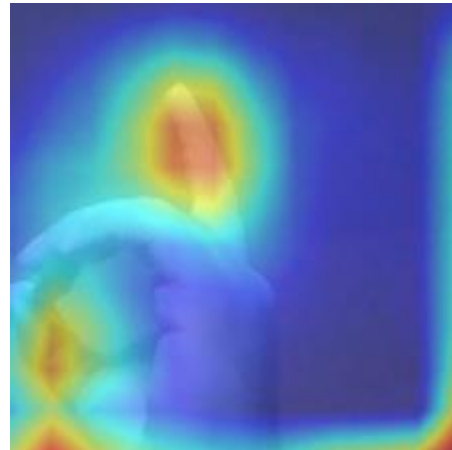
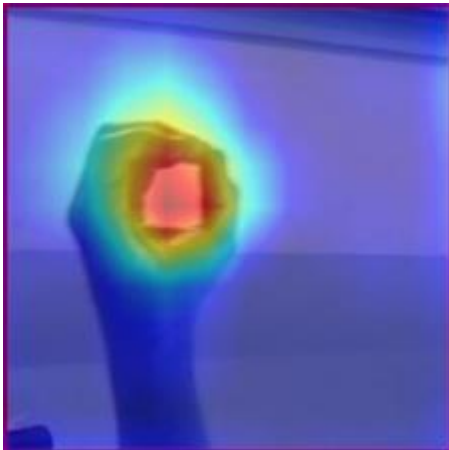
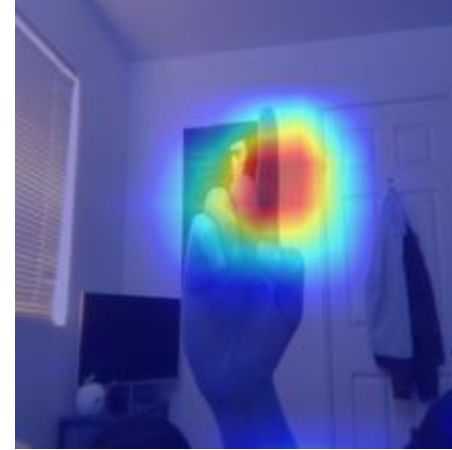
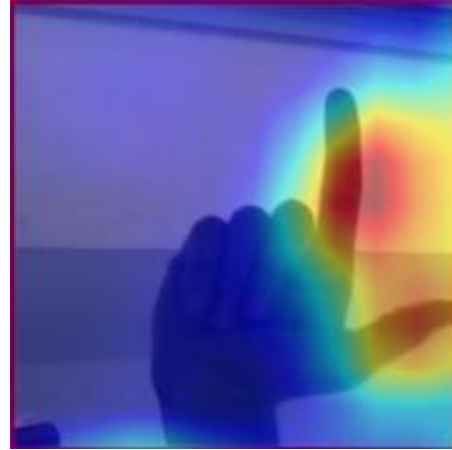
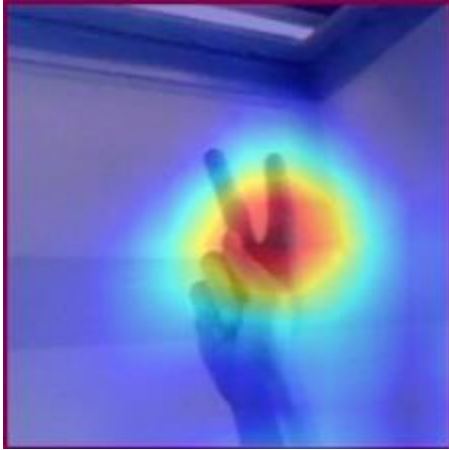
# Ds1 – AlexNet

Learning rate: 3.99e-05; Batch size: 64; Epochs: 35

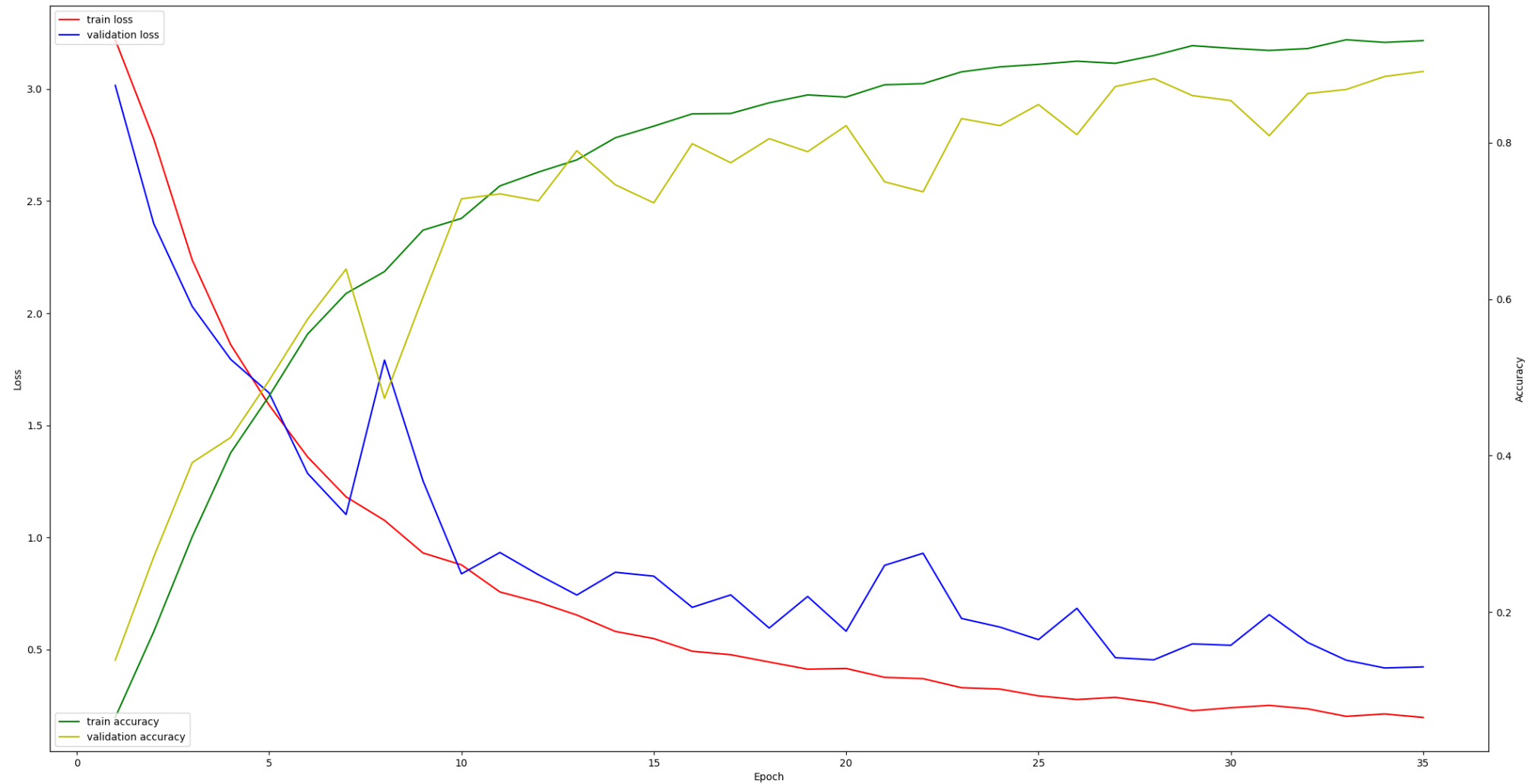




# Ds1 - AlexNet

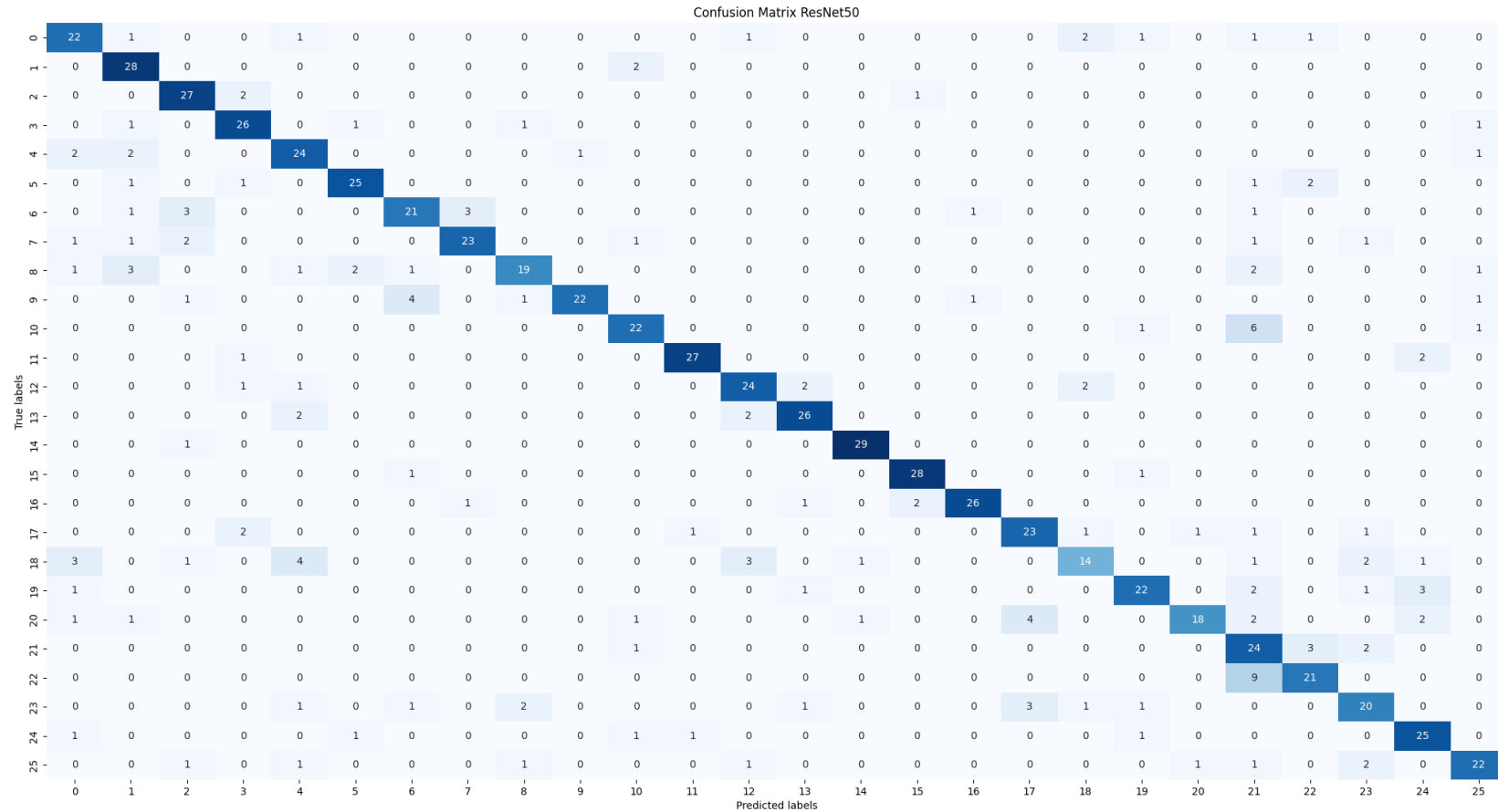


# Ds1 - AlexNet

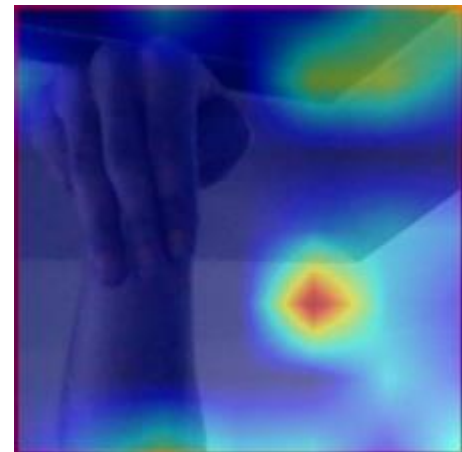
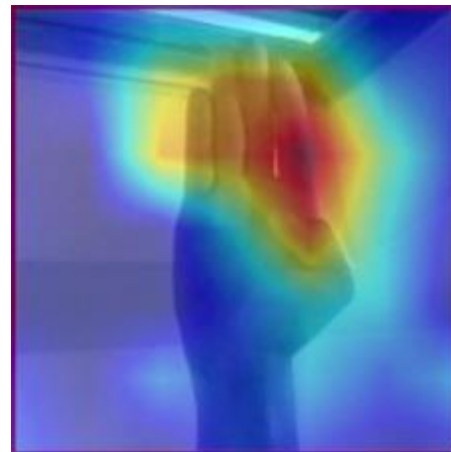
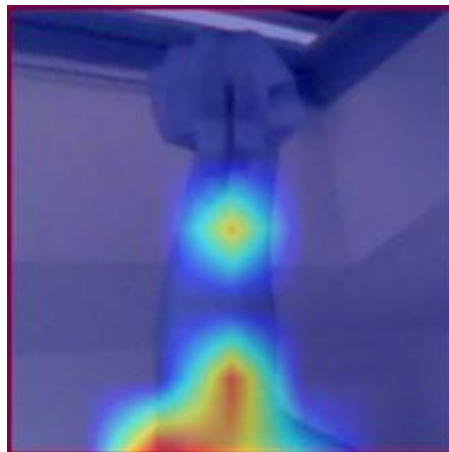
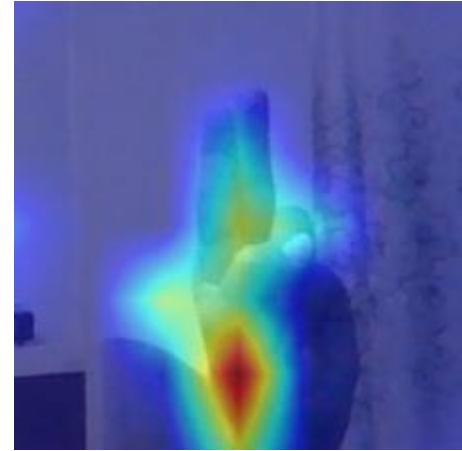
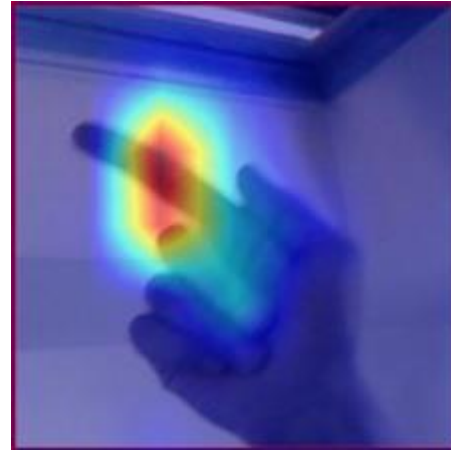
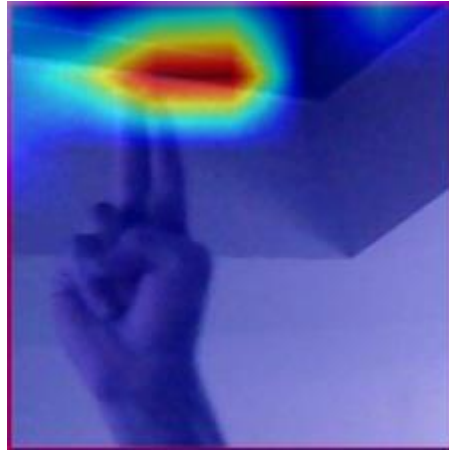
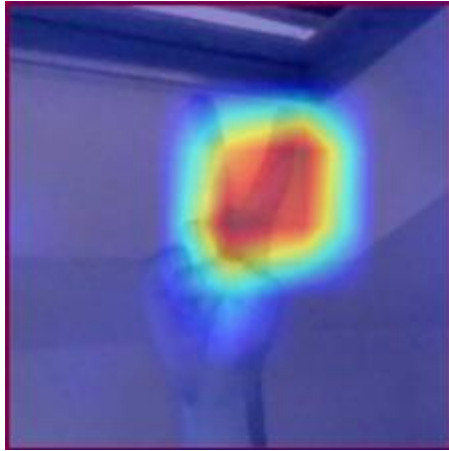


# Ds1 – ResNet50

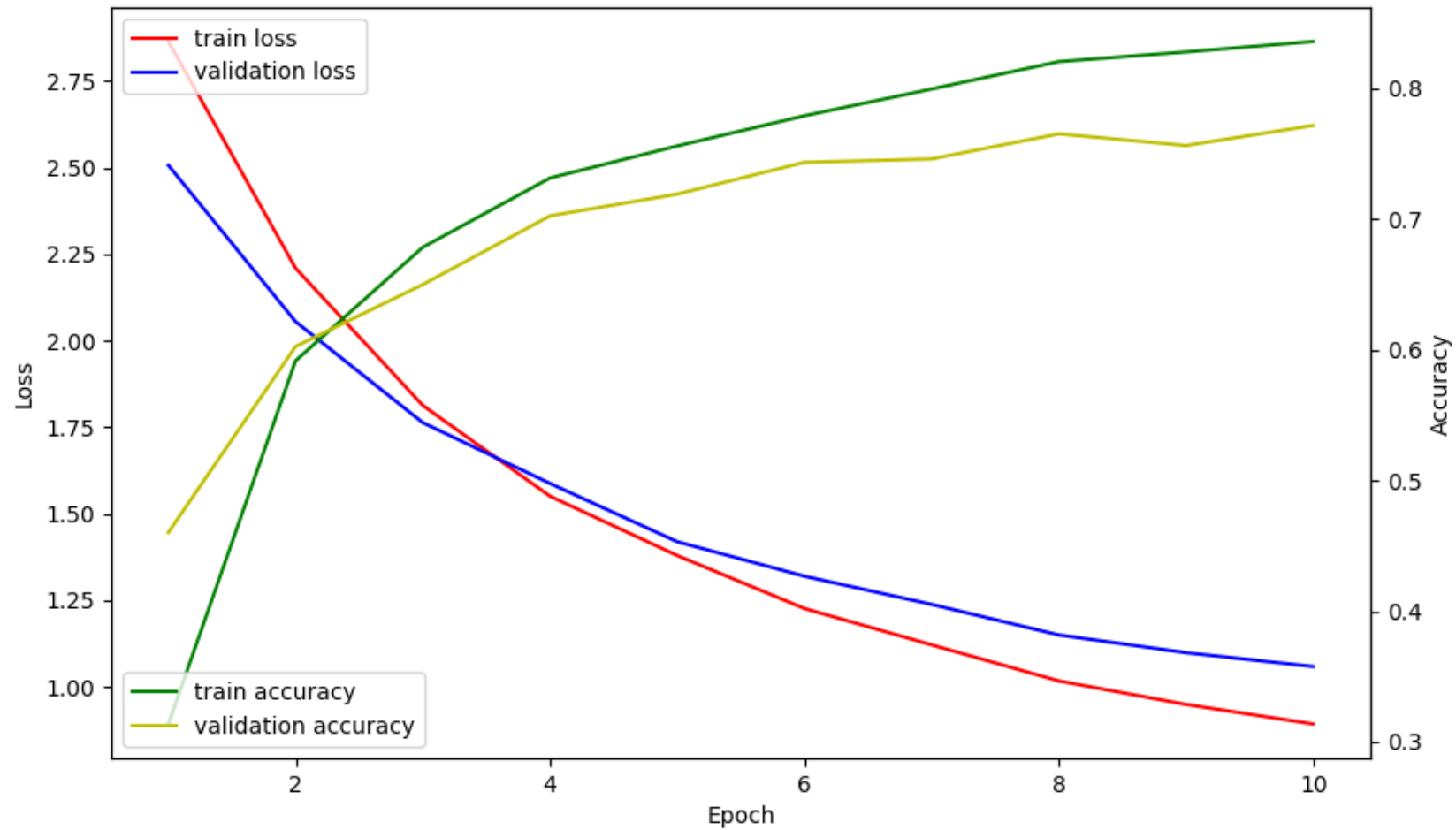
Learning rate: 0.0008; Batch size: 64; Epochs: 10



# Ds1 – ResNet50

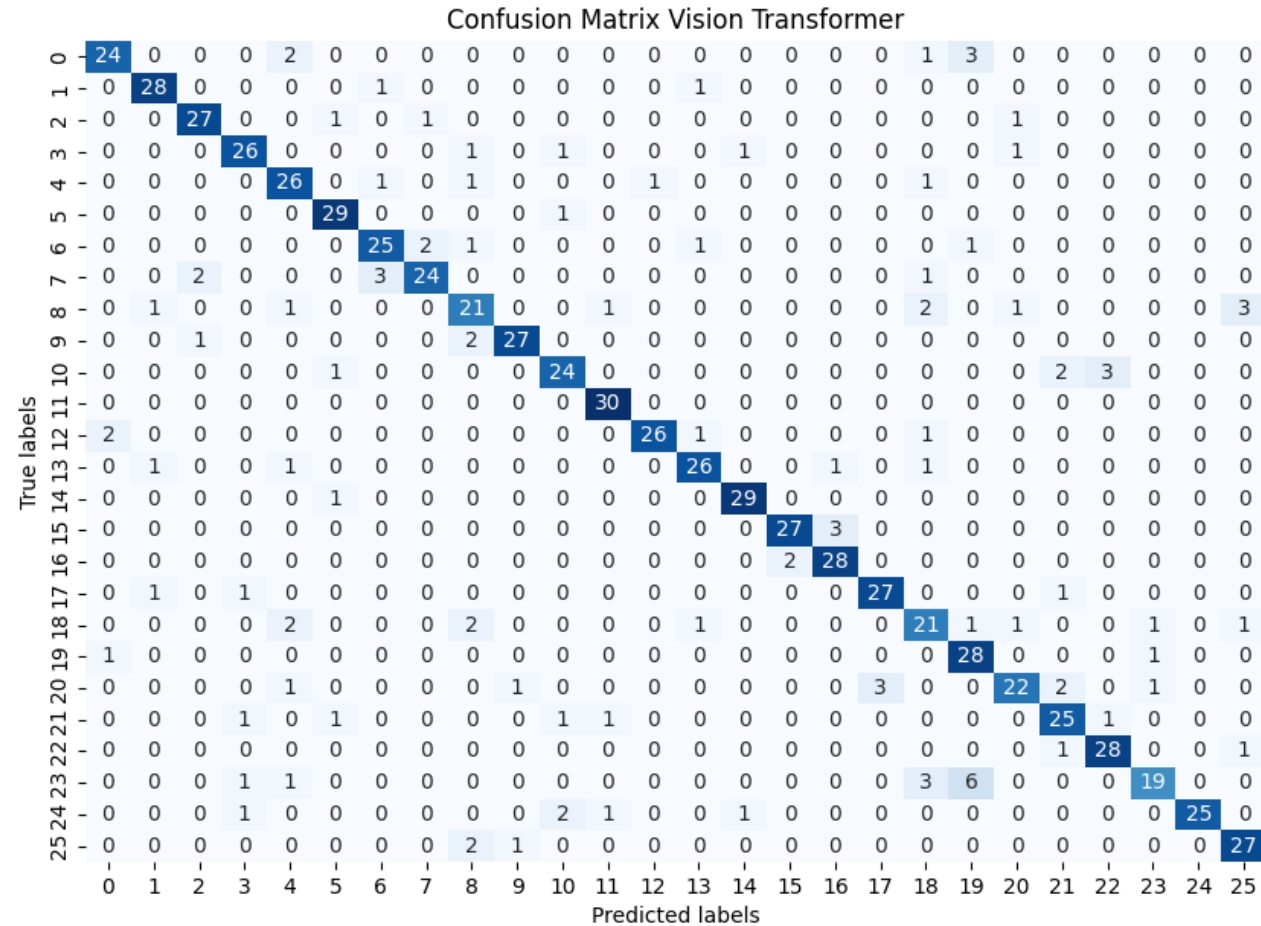


# Ds1 – ResNet50



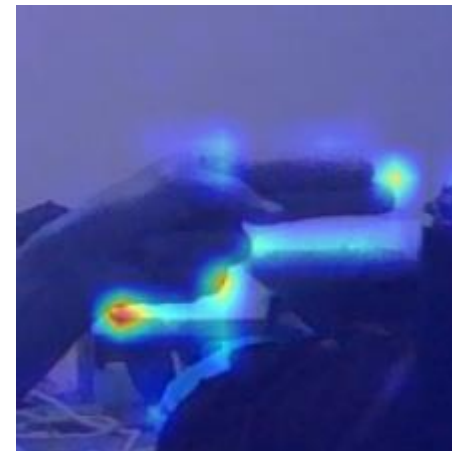
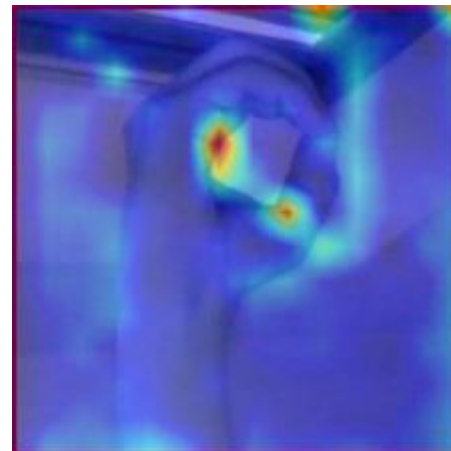
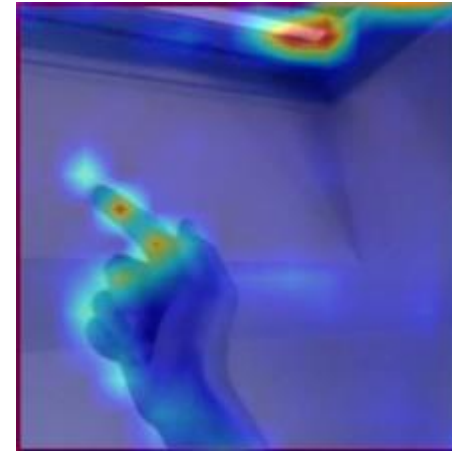
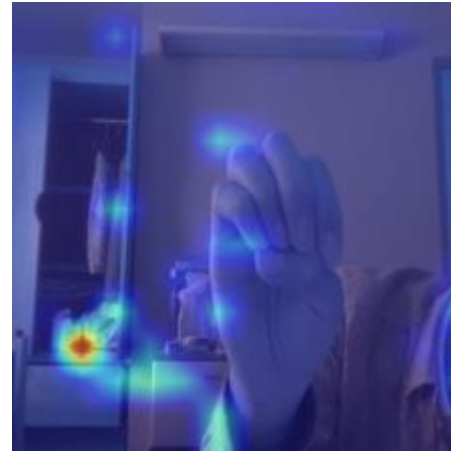
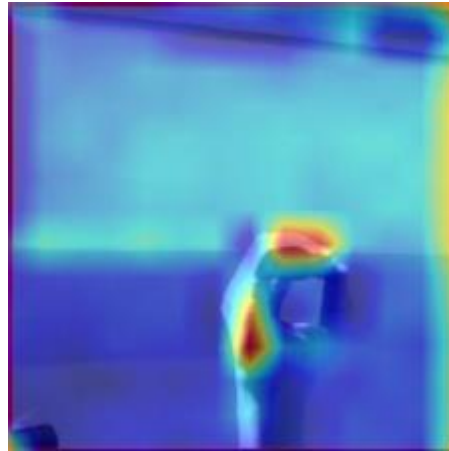
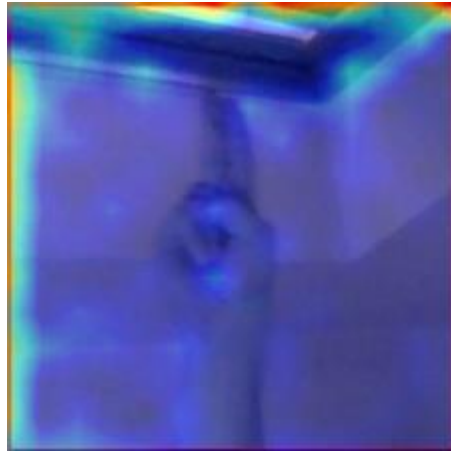
# Ds1 – VisionTransformer

Learning rate: 0.0008; Batch size: 64; Epochs: 10

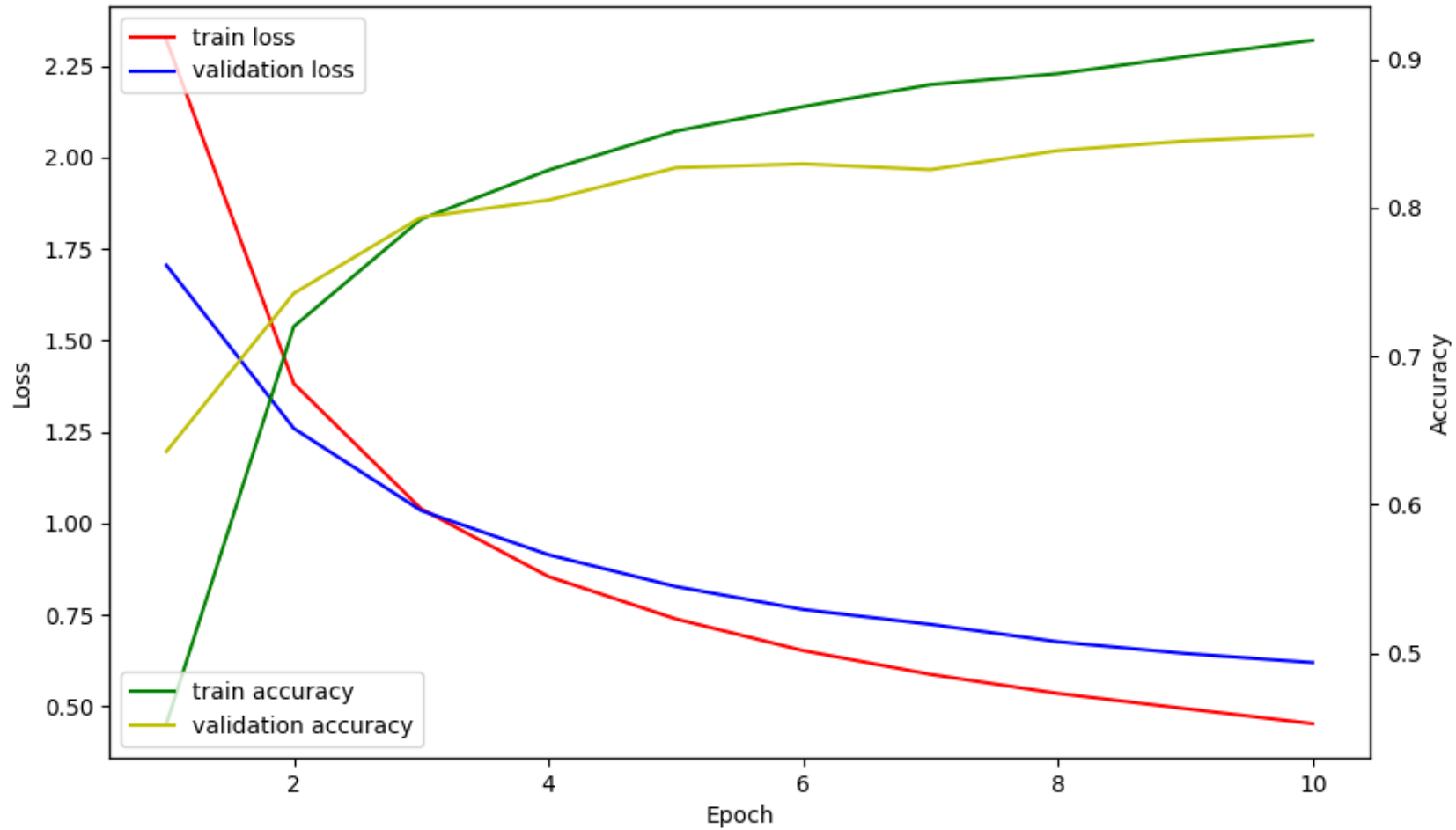




# Ds1 – VisionTransformer



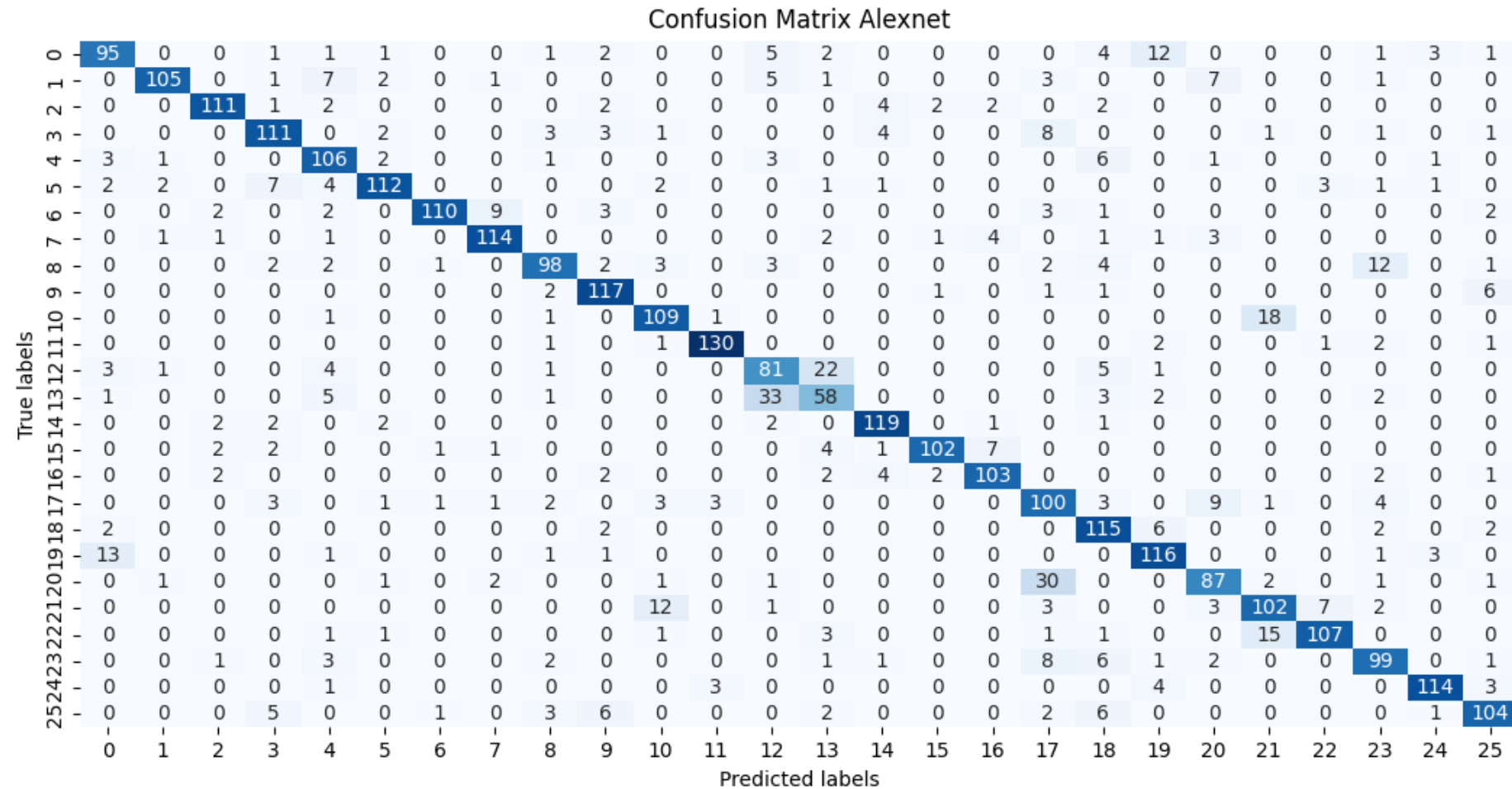
# Ds1 – VisionTransformer



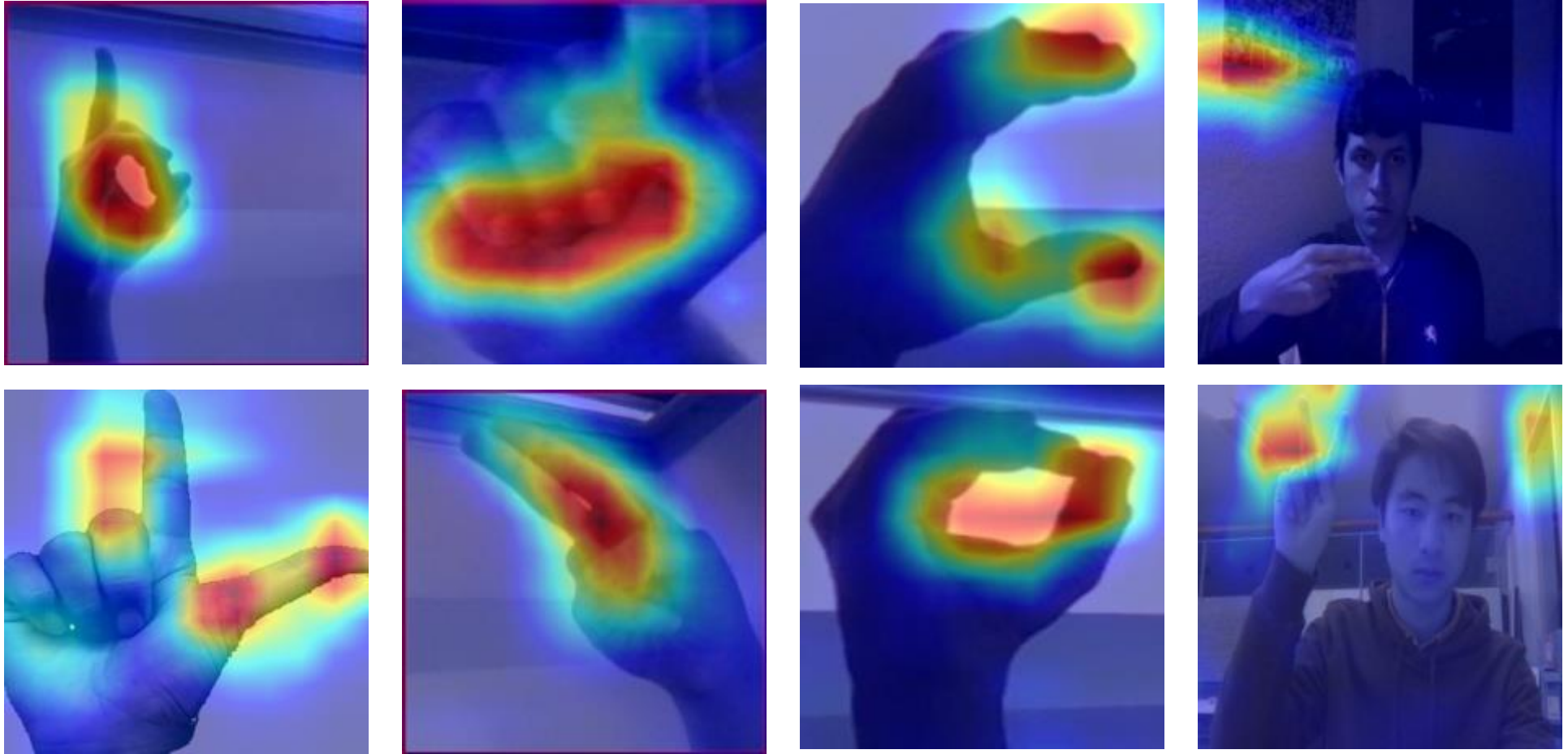


# Ds2 – AlexNet

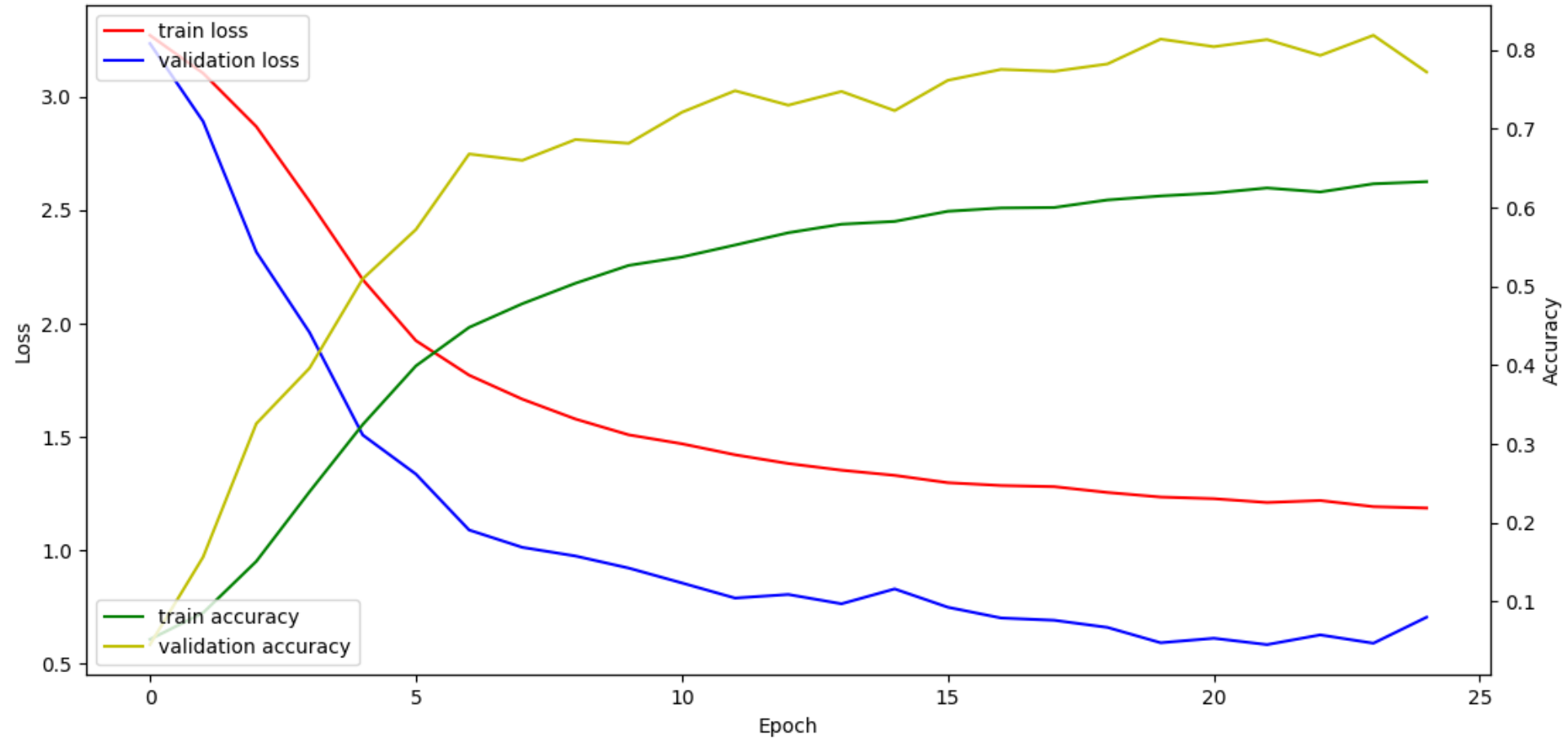
Batch size: 64, Drop out: 0.5, Learning rate: 0.0008, Epochs: 50 (early stopping)



# Ds2 – AlexNet



# Ds2 – AlexNet



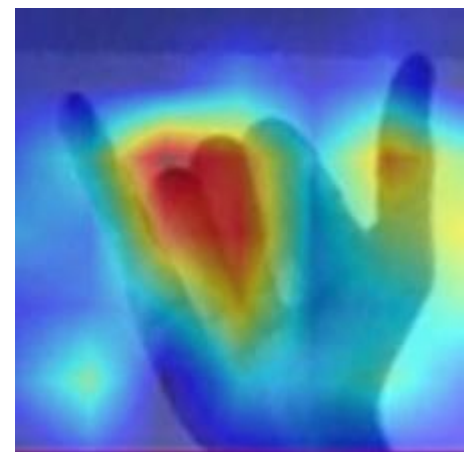
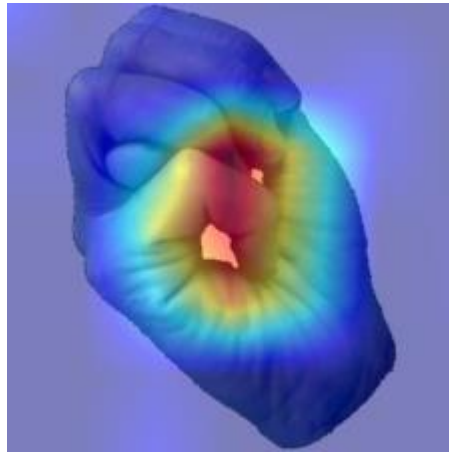
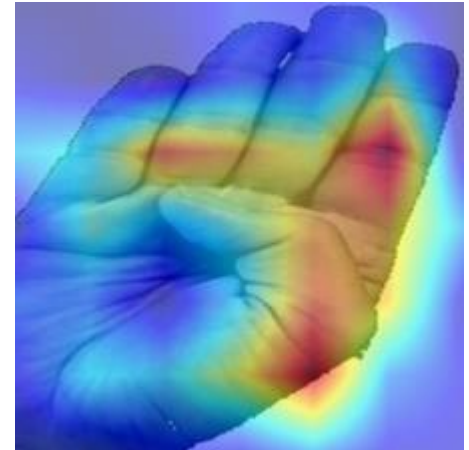
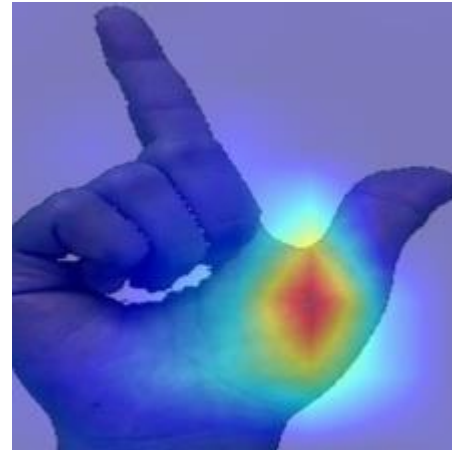
# Ds2 – ResNet50

Batch size: 64, Drop out: 0.2, Learning rate: 0.001, Epochs: 50 (early stopping)

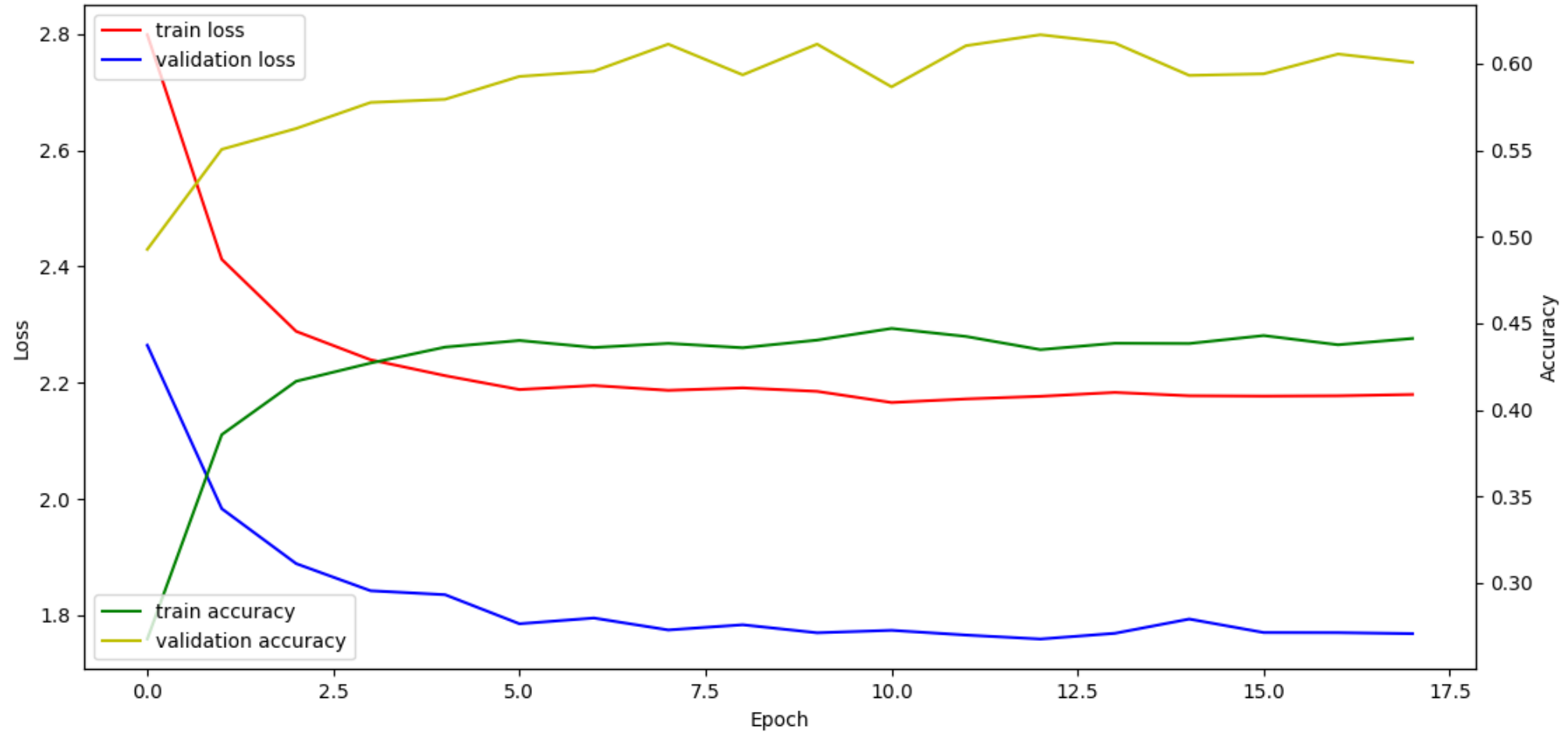
Confusion Matrix ResNet50

True labels	Predicted labels																									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	72	5	1	1	18	1	4	0	1	0	0	0	3	0	0	0	2	1	12	0	1	1	3	3	0	0
1	1	91	0	2	7	2	1	2	0	0	1	0	0	1	0	0	0	5	0	3	5	0	6	3	3	0
2	0	0	85	0	2	4	1	2	0	1	1	1	0	1	1	4	17	3	0	0	0	1	1	0	1	0
3	0	2	1	97	1	7	2	2	1	2	0	1	0	0	0	1	3	2	0	1	3	4	2	1	2	0
4	8	6	0	2	82	3	0	1	1	1	3	1	0	2	1	0	1	1	3	4	3	1	0	0	0	0
5	1	8	1	4	2	91	2	0	1	0	4	2	0	1	0	0	0	1	0	3	3	0	6	0	6	0
6	1	2	2	0	1	1	91	4	0	5	1	4	0	3	0	5	0	0	0	2	0	0	3	0	5	2
7	0	1	1	0	2	1	11	97	0	2	0	0	0	5	1	0	1	0	0	0	1	0	3	1	2	0
8	2	1	1	3	8	0	4	1	44	8	1	3	0	3	0	0	0	10	0	3	5	11	6	9	5	2
9	0	0	0	0	2	2	1	4	1	85	1	2	0	0	0	1	3	1	1	0	0	0	6	1	13	4
10	0	2	0	3	0	0	4	0	1	1	57	2	0	0	0	0	0	9	0	3	4	13	27	0	4	0
11	0	0	0	4	1	2	2	0	0	1	1	97	0	0	0	1	4	2	0	2	0	3	3	2	12	1
12	2	1	0	4	10	2	5	2	2	1	1	0	42	28	1	0	0	2	0	5	3	0	2	1	4	0
13	0	0	0	0	5	0	1	1	1	1	0	1	4	77	0	0	0	1	3	3	1	0	2	0	4	0
14	0	2	5	13	3	4	0	2	0	0	0	0	0	1	82	0	2	3	0	0	3	2	4	1	1	1
15	0	0	0	0	2	1	1	0	0	0	0	0	0	1	0	102	11	0	0	0	0	0	1	1	0	0
16	0	2	7	0	2	0	0	0	0	0	0	0	0	1	0	14	89	0	0	0	0	0	2	0	0	1
17	1	2	0	2	0	1	3	0	2	1	3	0	2	0	0	0	0	88	1	1	11	2	2	4	5	0
18	2	0	0	2	16	0	0	0	0	1	2	0	0	1	1	1	0	2	69	11	4	1	0	12	2	2
19	4	0	0	1	9	2	0	0	0	0	1	6	0	2	0	1	0	0	1	75	1	2	3	5	21	2
20	0	4	0	4	6	2	0	2	1	0	2	0	0	2	0	0	0	14	0	1	60	11	3	11	4	0
21	0	1	0	1	1	7	0	0	0	1	10	2	0	0	0	0	0	5	0	1	3	70	21	5	2	0
22	0	1	0	0	2	3	0	2	0	0	3	0	0	0	0	0	0	3	0	1	4	12	97	0	2	0
23	2	0	1	1	13	0	1	0	2	2	0	0	0	0	0	0	0	5	4	7	11	9	0	61	5	1
24	0	0	0	0	1	2	3	0	0	3	0	13	0	1	0	0	0	0	0	4	0	2	11	0	83	2
25	0	0	0	0	2	0	7	2	0	9	1	2	0	1	0	1	2	2	5	9	5	9	2	1	22	48

# Ds2 – ResNet50



# Ds2 – ResNet50





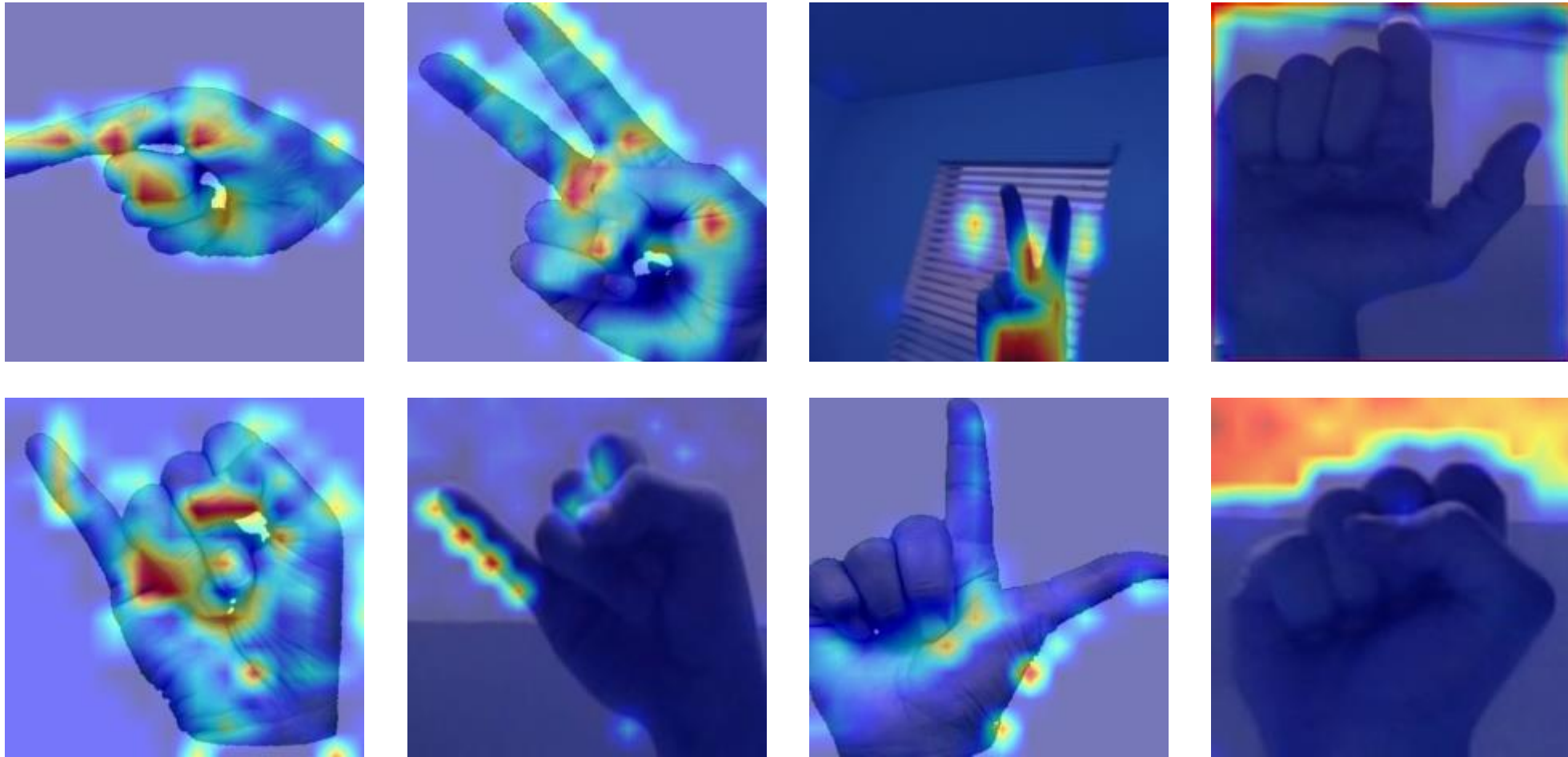
# Ds2 – VisionTransformer

Batch size: 64, Drop out: 0.3, Learning rate: 0.0001, Epochs: 50

Confusion Matrix Vision Transformer

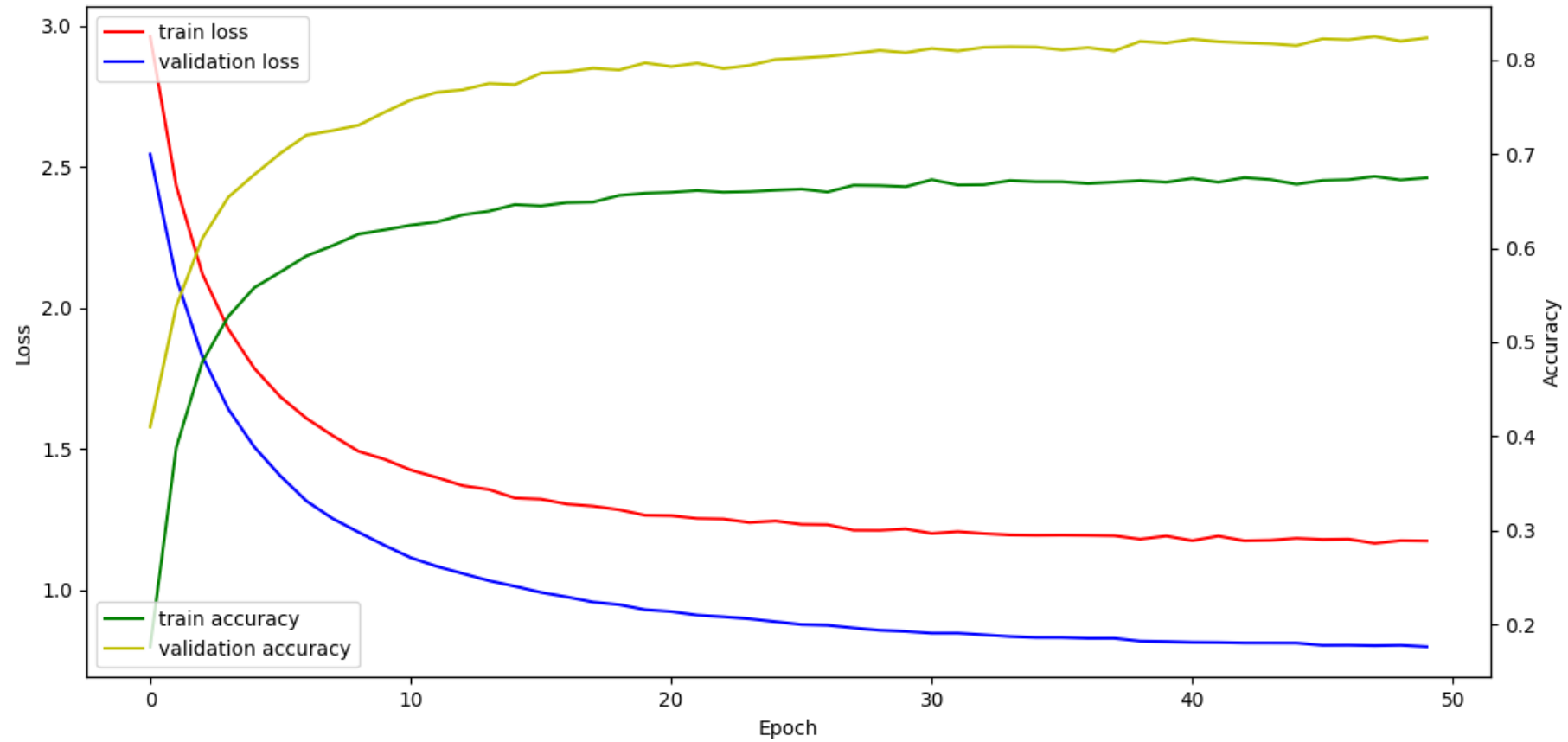
True labels	Predicted labels																									
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
0	94	1	1	0	5	1	1	0	3	1	0	0	2	3	0	0	0	0	6	8	0	0	0	0	2	1
1	0	119	0	0	2	6	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	1	0
2	0	0	111	0	0	2	3	0	0	3	0	0	0	0	1	0	5	1	0	0	0	0	0	0	0	0
3	1	0	9	104	0	4	0	0	7	3	0	0	1	1	4	0	0	0	0	0	0	0	0	0	1	0
4	6	0	1	1	97	0	0	1	0	1	0	0	2	6	0	0	0	0	4	1	0	0	0	1	2	1
5	1	2	0	0	0	126	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	2	0	1	0
6	0	0	0	1	0	0	115	8	0	3	0	0	0	0	0	1	0	1	0	0	0	0	0	3	0	0
7	0	0	0	1	0	5	3	119	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
8	3	1	0	1	3	2	0	0	93	8	0	4	0	1	0	1	0	1	4	0	2	1	0	3	2	0
9	0	0	1	2	1	1	2	1	1	116	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1
10	0	0	0	0	0	7	1	0	2	2	103	0	0	0	0	0	0	0	0	1	0	11	2	0	1	0
11	0	0	0	0	0	3	0	0	1	1	0	122	0	1	0	0	0	2	0	0	0	0	0	0	8	0
12	3	0	0	0	2	0	1	1	0	0	0	0	88	16	3	0	1	0	1	0	0	1	0	0	1	0
13	1	0	0	0	0	0	0	1	0	0	0	0	9	91	0	0	1	0	2	0	0	0	0	0	0	0
14	0	1	1	0	0	0	0	3	0	0	0	0	0	1	123	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	116	2	0	0	0	0	0	0	0	0
16	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	3	111	0	0	0	0	0	0	0	0
17	0	0	0	0	2	2	0	4	5	2	2	0	0	0	0	0	0	0	102	0	0	12	0	0	0	0
18	2	0	0	0	8	0	0	0	4	0	0	0	1	0	1	0	0	0	0	107	4	0	0	0	1	0
19	4	0	0	0	1	0	0	0	0	0	1	3	3	4	0	0	0	0	5	98	0	0	0	3	13	1
20	0	0	0	0	1	5	0	3	3	2	1	0	0	0	0	0	0	5	1	1	103	0	0	0	2	0
21	0	0	0	2	0	3	1	0	4	0	10	2	0	1	0	0	0	2	0	0	7	86	6	3	3	0
22	0	1	0	1	0	2	0	0	0	0	1	1	0	1	0	0	0	1	0	0	0	2	111	0	9	0
23	0	0	0	0	4	2	1	1	1	0	0	0	1	1	0	0	0	1	12	6	2	0	0	87	3	3
24	0	0	0	0	0	4	0	0	1	1	2	2	0	0	0	0	0	0	0	2	0	0	0	0	113	0
25	0	0	0	1	0	0	2	1	8	9	0	0	0	0	0	0	0	0	3	1	0	0	0	0	1	104

# Ds2 – VisionTransformer





# Ds2 – VisionTransformer



# Conclusion

# Conclusion

- **Best Models:** Vision Transformer, AlexNet
- **Data Augmentation:** More images and augmentation improve model performance
- **Early Stopping:** Adjusting patience prevents overfitting and enhances generalization
- **Optimization:** Dropout, weight decay, (learning rate scheduling) improve robustness

# Outlook

# Outlook

- Fine-tuning Diffusion model
- Optimize Prompt Engineering
- Improve data quality
- More experiments with hyperparameters