
# Bimodal Deep Learning Architecture for Malware Classification

Deniz Aytemiz

Committee members: Creed Jones, Angelos Stavrou, Kendall Giles

05/09/2023

# Presentation Outline:

1. Introduction
2. The Deep Learning Models
    a. Image Classification Modality
        - CNN A model
    b. Natural Language Processing Modality
        - CNN B model
        - CNN C model
    c. Bimodal Architecture
        - Sequential Model
3. Experimental Results & Discussion
4. Conclusion

# Introduction

# Introduction: Malware Types

Adware

Spyware

Virus

Worm

Trojan

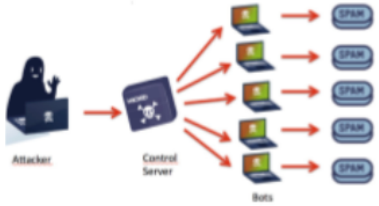
Malware

Rootkit


Backdoors

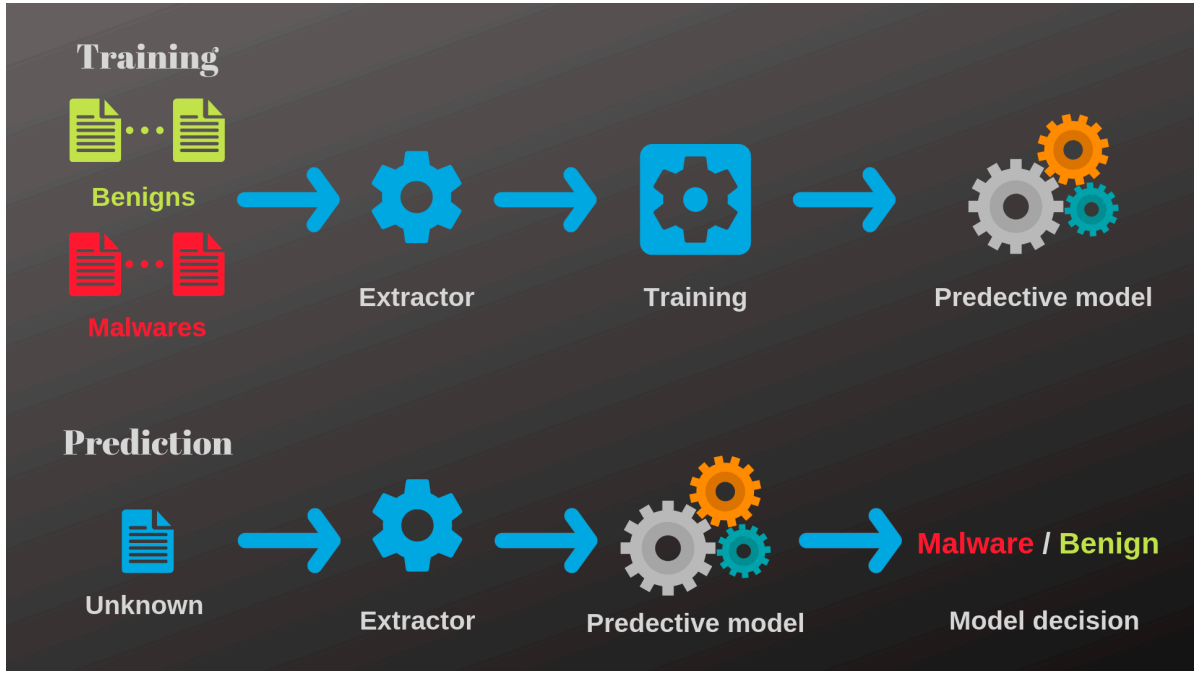
Ransomware


Command & Control Bot



# Introduction: Malware Detection and Classification

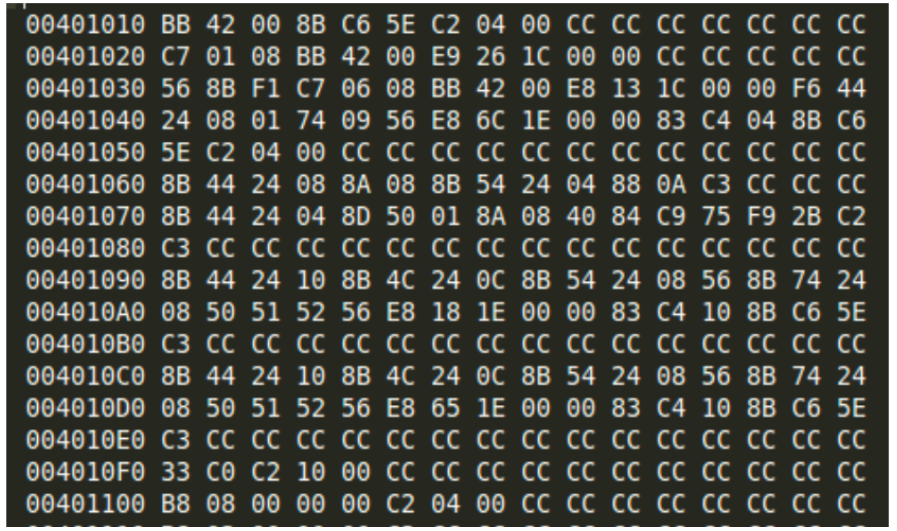
- Every day 560,000 new pieces of malware are detected.
- Malware develop polymorphic and metamorphic malware to evade detection.
- Conventional detection methods are insufficient to detect malware.
- Traditional antivirus solutions relies on signature-based and heuristic based methods.
- Use of machine learning techniques for malware classification and detection increased in the last years.
- Machine learning feature extraction of malware data;
  1. Static features, 2. Dynamic features



# Introduction: Objective

- The objective of this project is to classify malware samples from BIG2015 dataset into their corresponding malware families.

| index | name | # of instance in the dataset | type of malware |
|---|---|---|---|
| 1 | Ramnit | 1541 | Worm |
| 2 | Lollipop | 2478 | Adware |
| 3 | Kelihos_ver3 | 2942 | Backdoor |
| 4 | Vundo | 475 | Trojan |
| 5 | Simda | 42 | Backdoor |
| 6 | Tracur | 751 | TrojanDownloader |
| 7 | Kelihos_ver1 | 398 | Backdoor |
| 8 | Obfuscator.ACY | 1228 | Obfuscated malware |
| 9 | Gatak | 1013 | Backdoor |

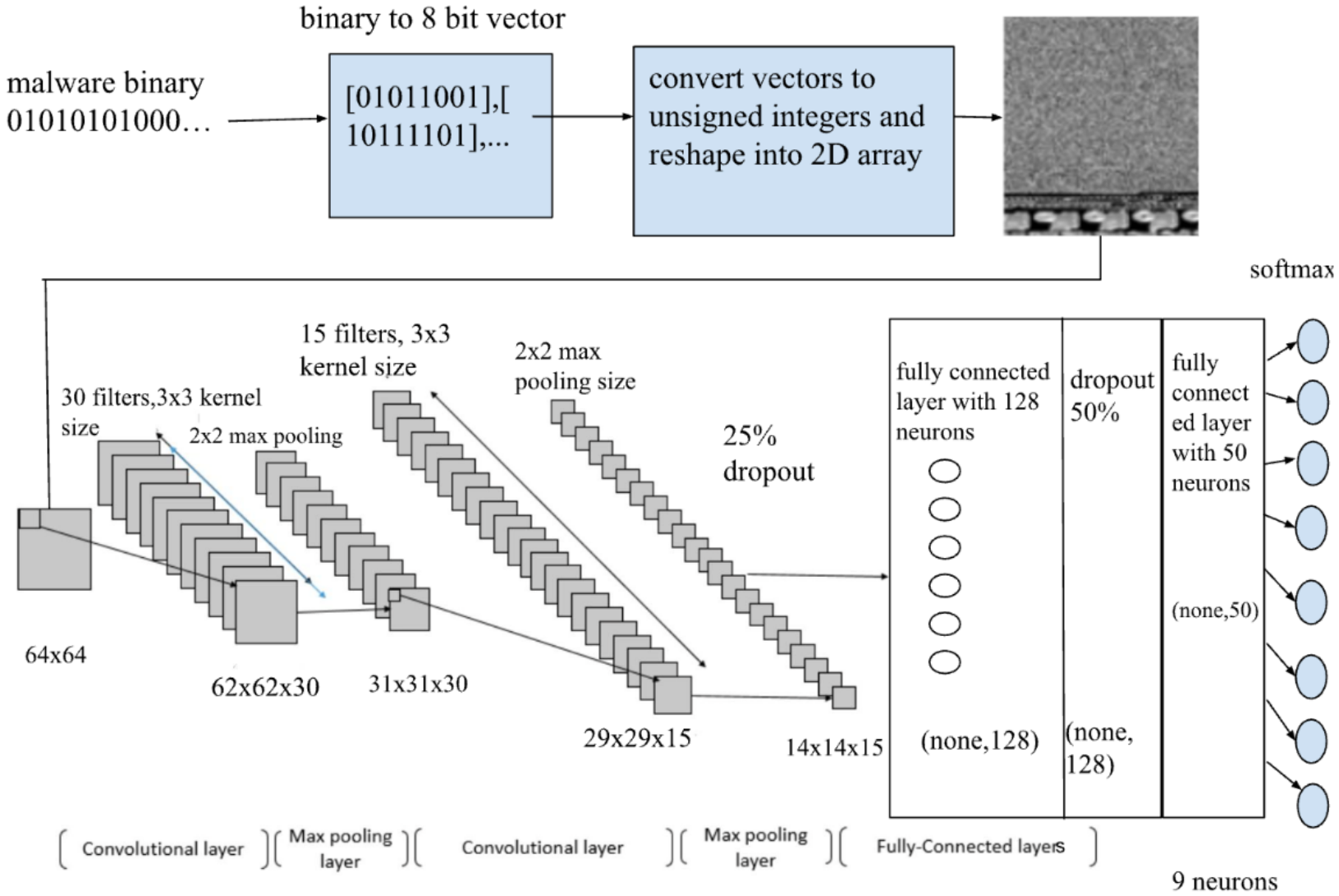



```
.text:00401174                                 ; ---------------------------------------
.text:00401177 CC CC CC CC CC CC CC CC CC       align 10h
.text:00401180 6A 04                           push    4
.text:00401182 68 00 10 00 00                  push    1000h
.text:00401187 68 68 BE 1C 00                  push    1CBE68h
.text:0040118C 6A 00                           push    0
.text:0040118E FF 15 9C 63 52 00               call    ds:GetCurrentProcess
.text:00401194 50                              push    eax
.text:00401195 FF 15 C8 63 52 00               call    ds:VirtualAllocEx
.text:0040119B 8B 4C 24 04                     mov     ecx, [esp+4]
.text:0040119F 6A 00                           push    0
.text:004011A1 6A 40                           push    40h
.text:004011A3 68 68 BE 1C 00                  push    1CBE68h
.text:004011A8 50                              push    eax
.text:004011A9 89 01                           mov     [ecx], eax
.text:004011AB FF 15 C4 63 52 00               call    ds:VirtualProtect
.text:004011B1 B8 04 00 00 00                  mov     eax, 4
.text:004011B6 C2 04 00                        retn    4
```

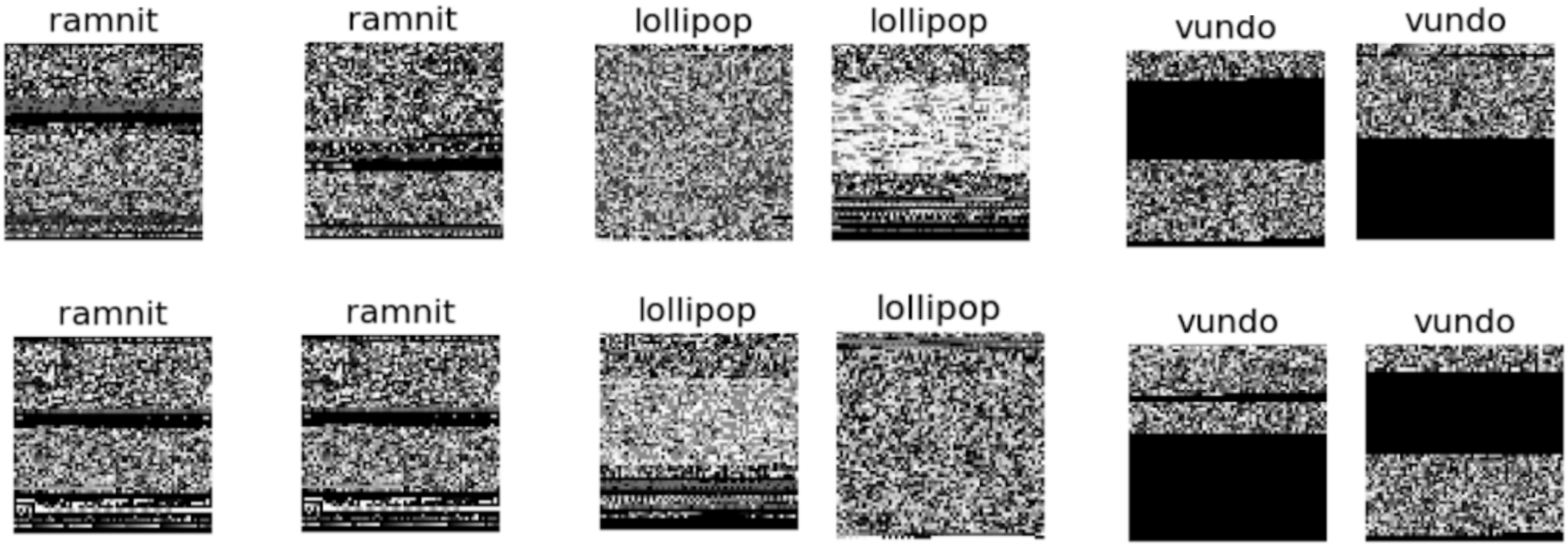
# The Deep Learning Models:

- Image Classification Modality
- Natural Language Processing Modality
- Bimodal Sequential Model

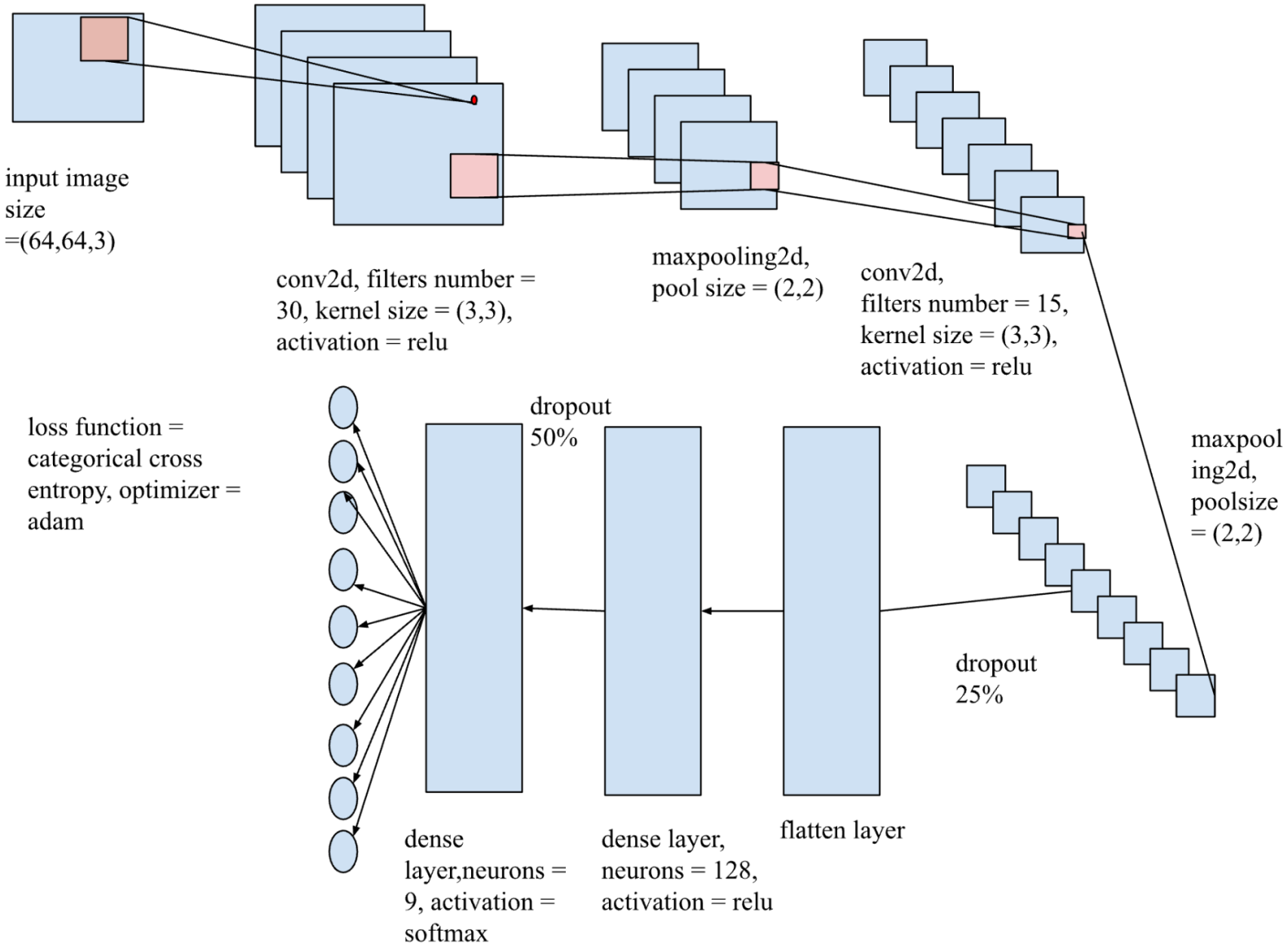
# Models: Image Classification Modality



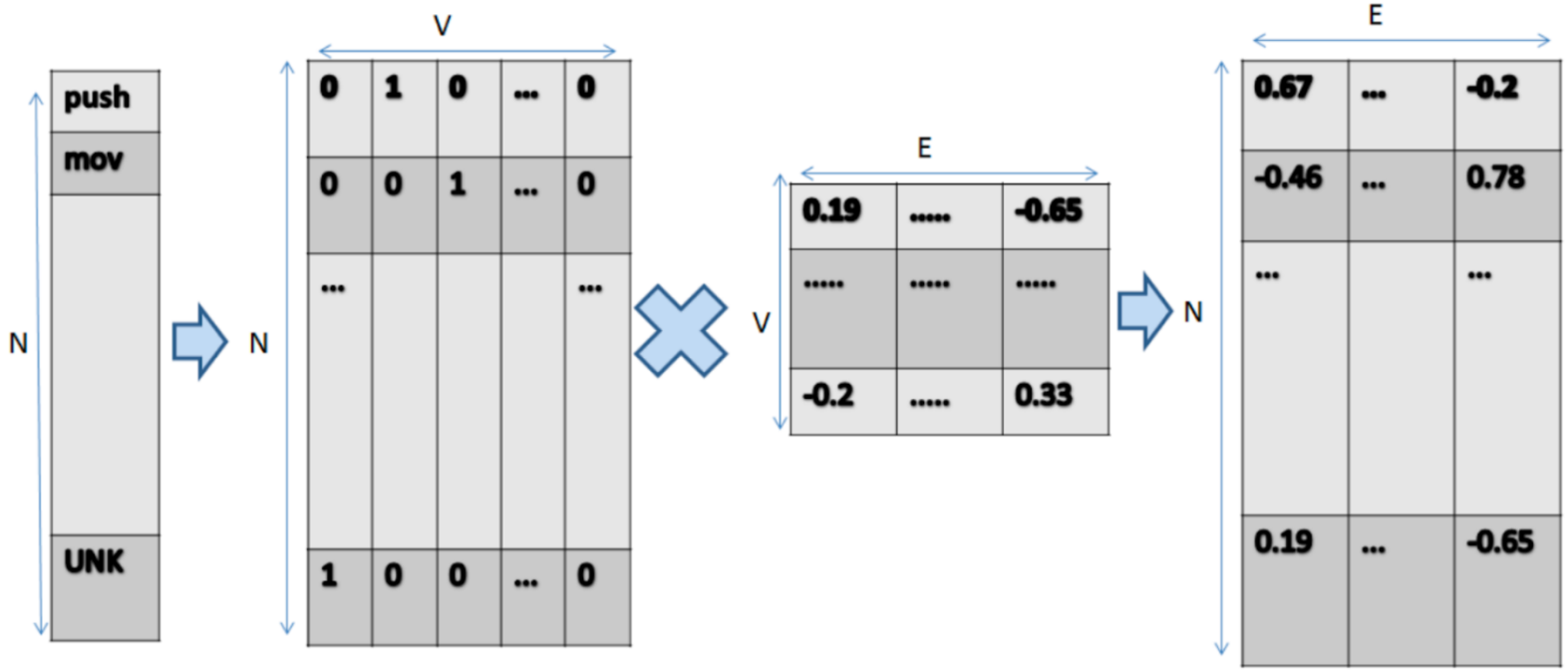
# Models: Raw Byte File to Grayscale Images



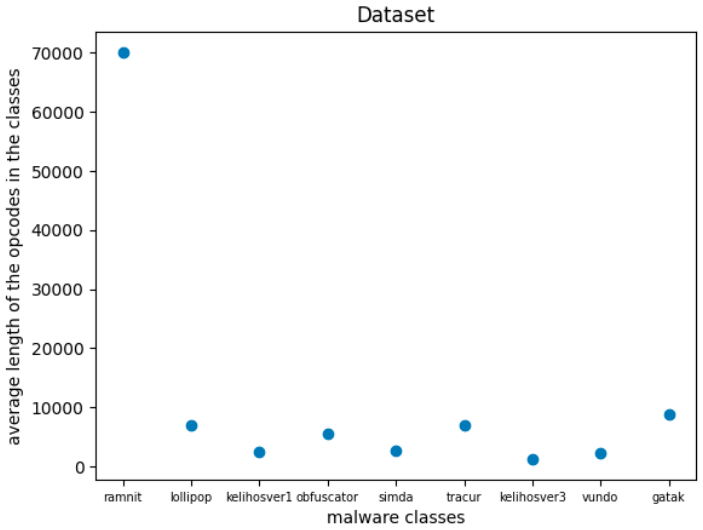
# Models: CNN A


input image size =(64,64,3)
conv2d, filters number = 30, kernel size = (3,3), activation = relu
maxpooling2d, pool size = (2,2)
conv2d, filters number = 15, kernel size = (3,3), activation = relu
maxpool ing2d, poolsize = (2,2)
dropout 25%
flatten layer
dense layer, neurons = 128, activation = relu
dropout 50%
dense layer,neurons = 9, activation = softmax
loss function = categorical cross entropy, optimizer = adam

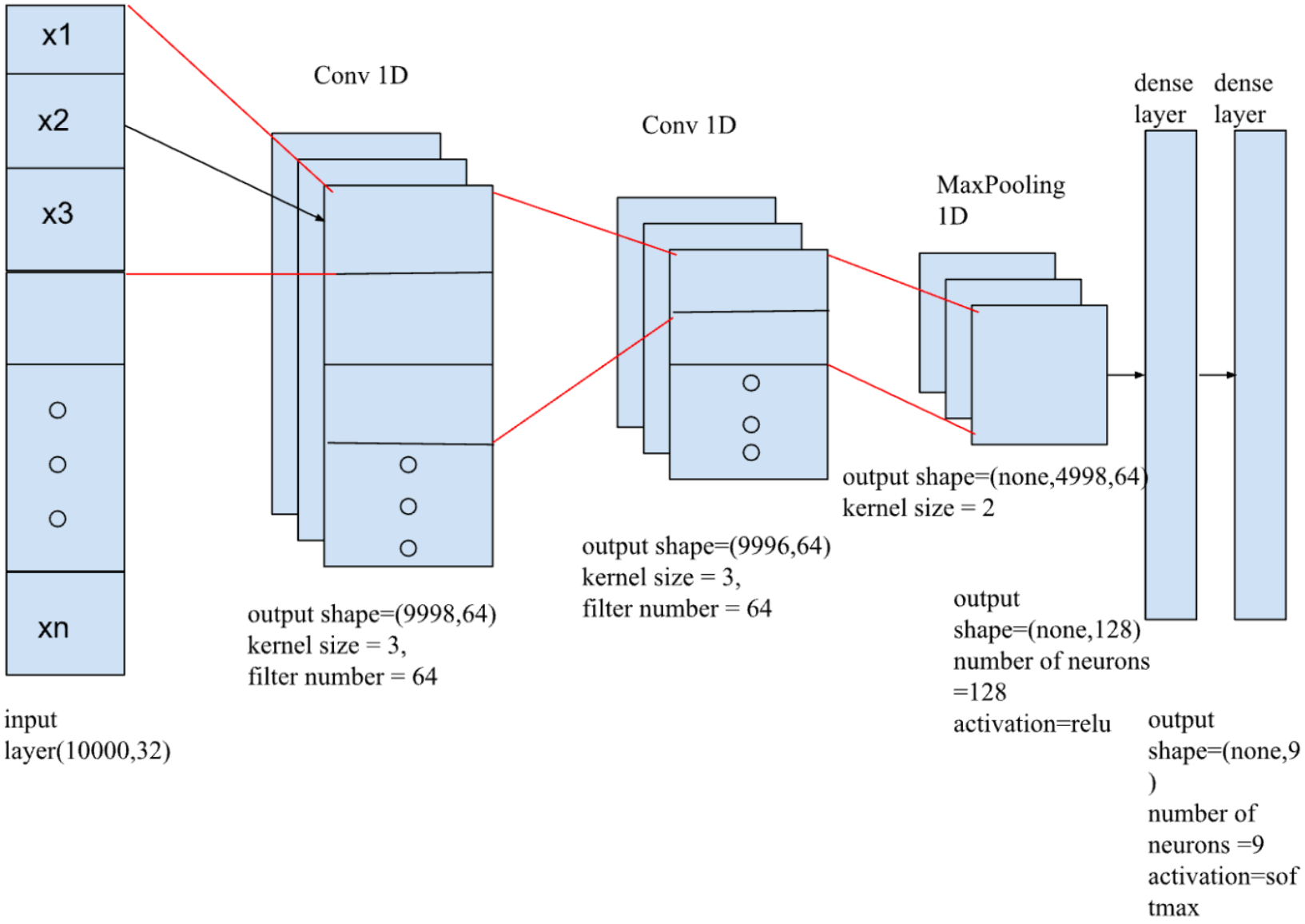
# Models: Natural Language Processing Modality- Word Embedding



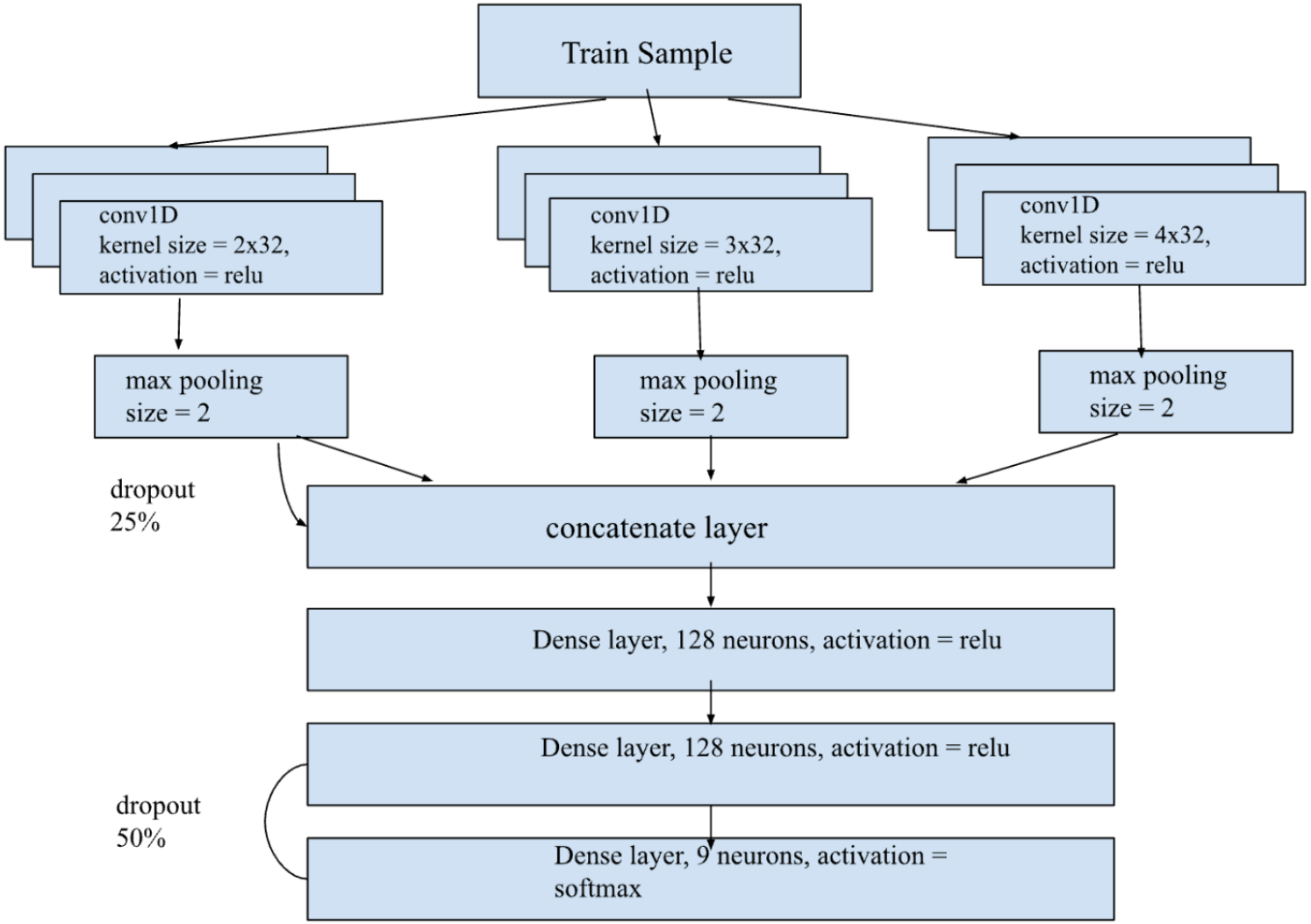

Predictive Language Model: Skip-grams
Word Embedding Size(E) = 32
Window Size = 5
Vocabulary Size(V) = 443
Sequence Length(N) = 10000



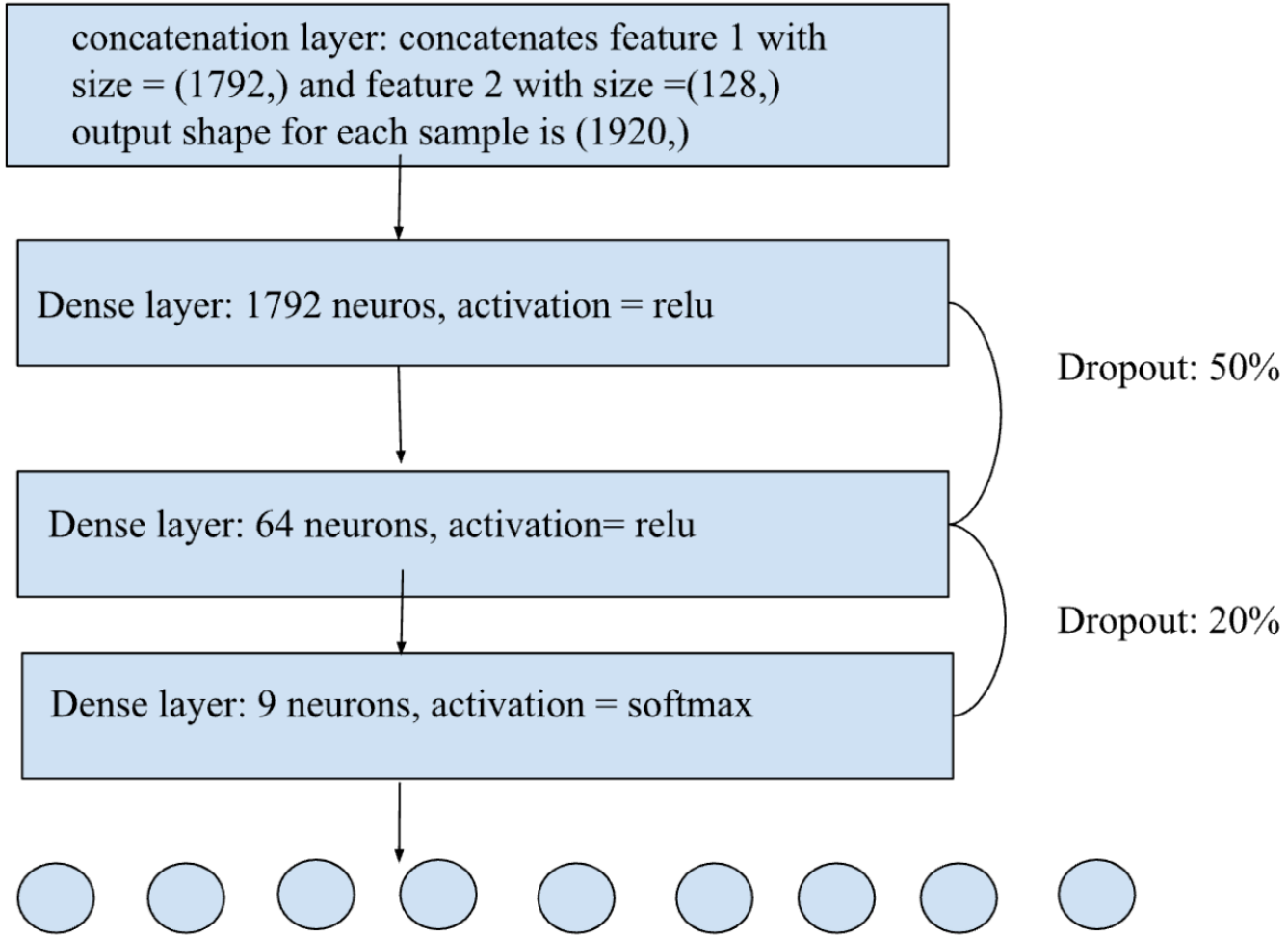
# Models: Natural Language Processing Modality - CNN B



# Models: Natural Language Processing Modality - CNN C



# Models: Bimodal Sequential Model

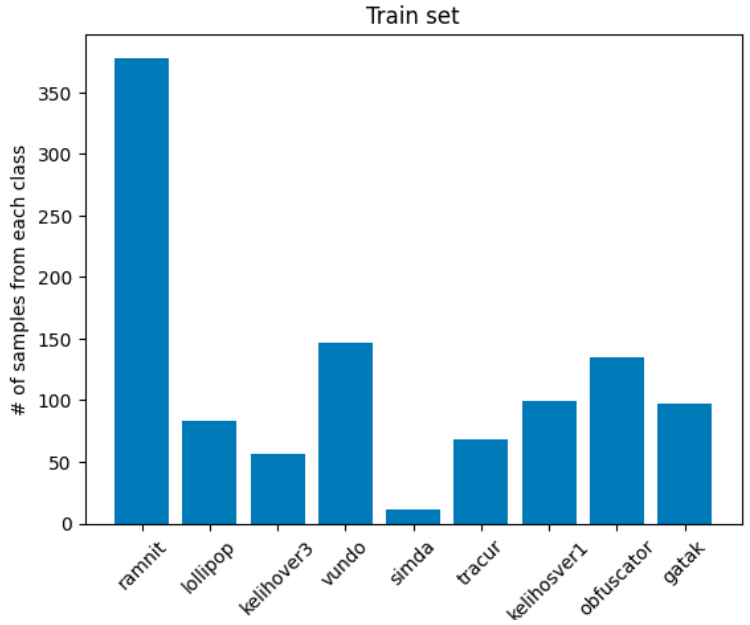


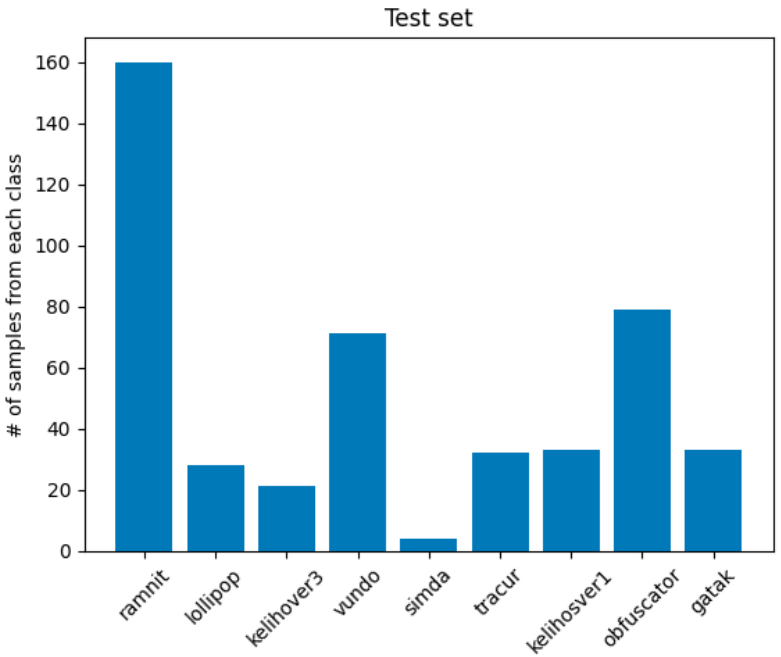

loss = categorical_crossentropy, optimizer = adam

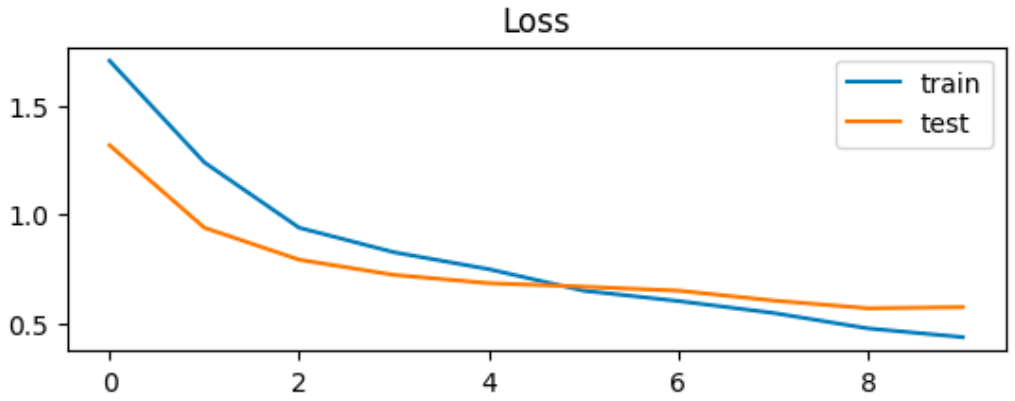
# Experimental Results &Discussion

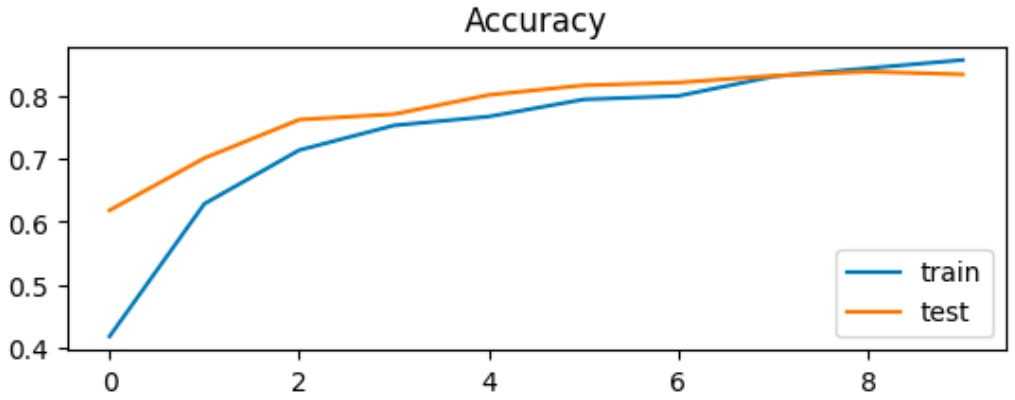
- Image Classification Modality
- Natural Language Processing Modality
- Bimodal Sequential Model

# Results: Image Classification Modality-CNN A



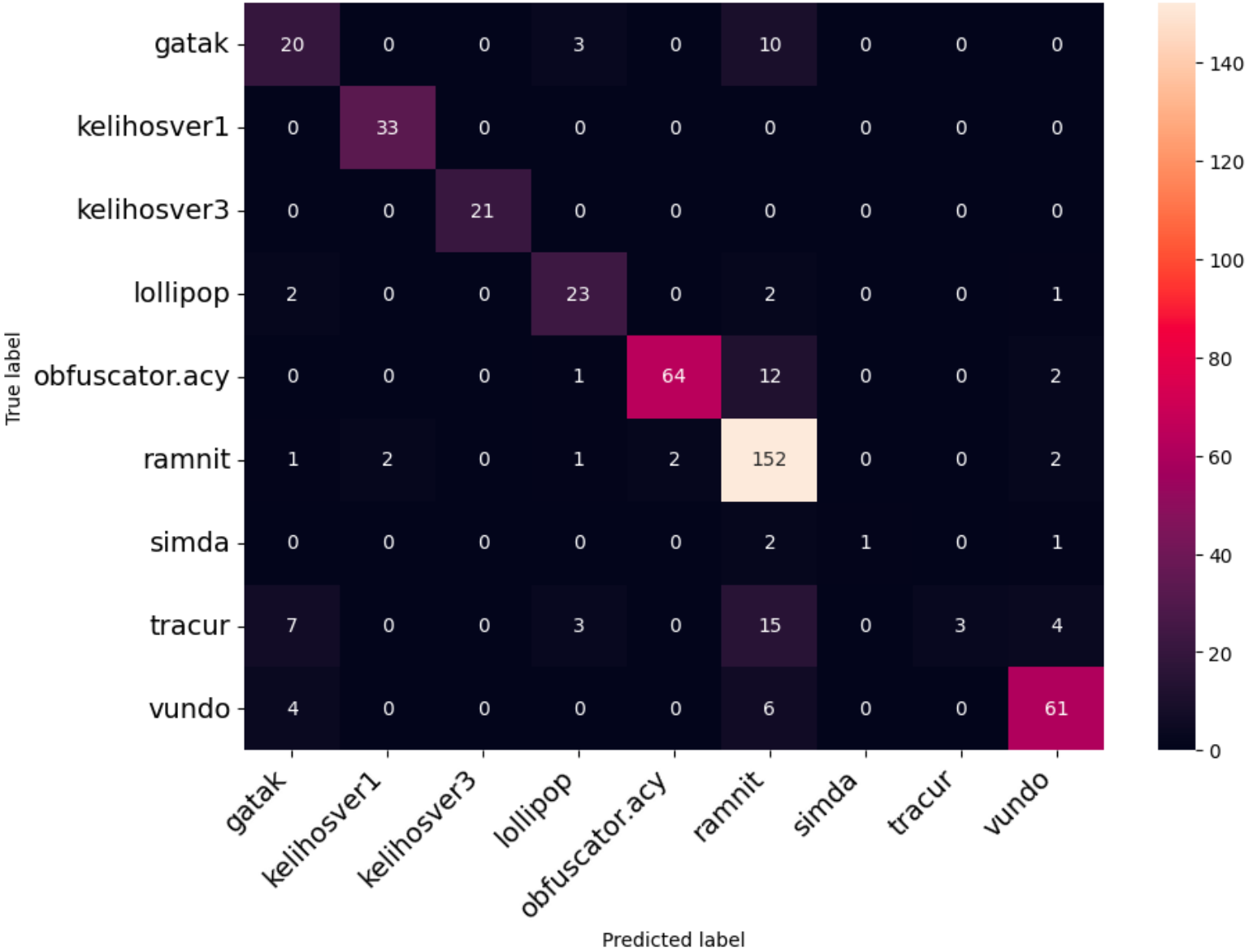




Test Accuracy: 0.8120

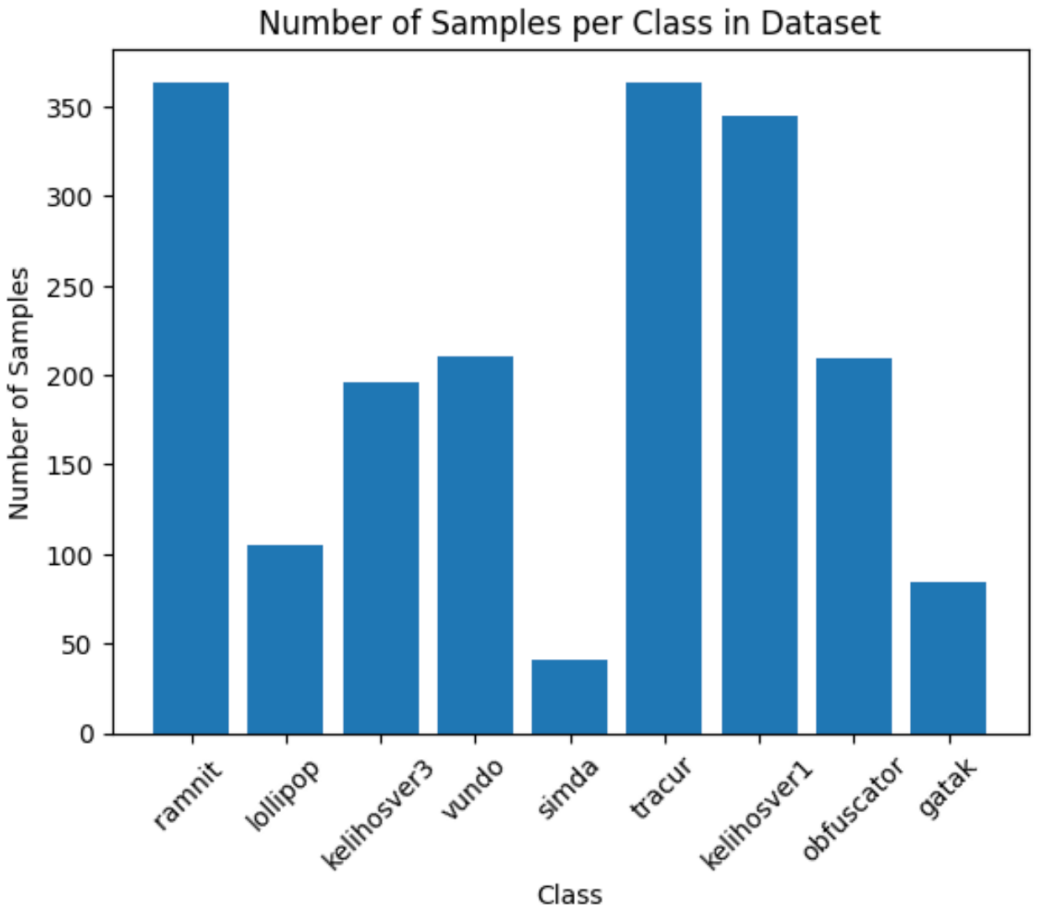
Google Collab environment with 25 RAM and 166.77 GB disk capacity. GPU and TPU runtimes are up to 12 hours and the GPUs are K80, P100, T4.

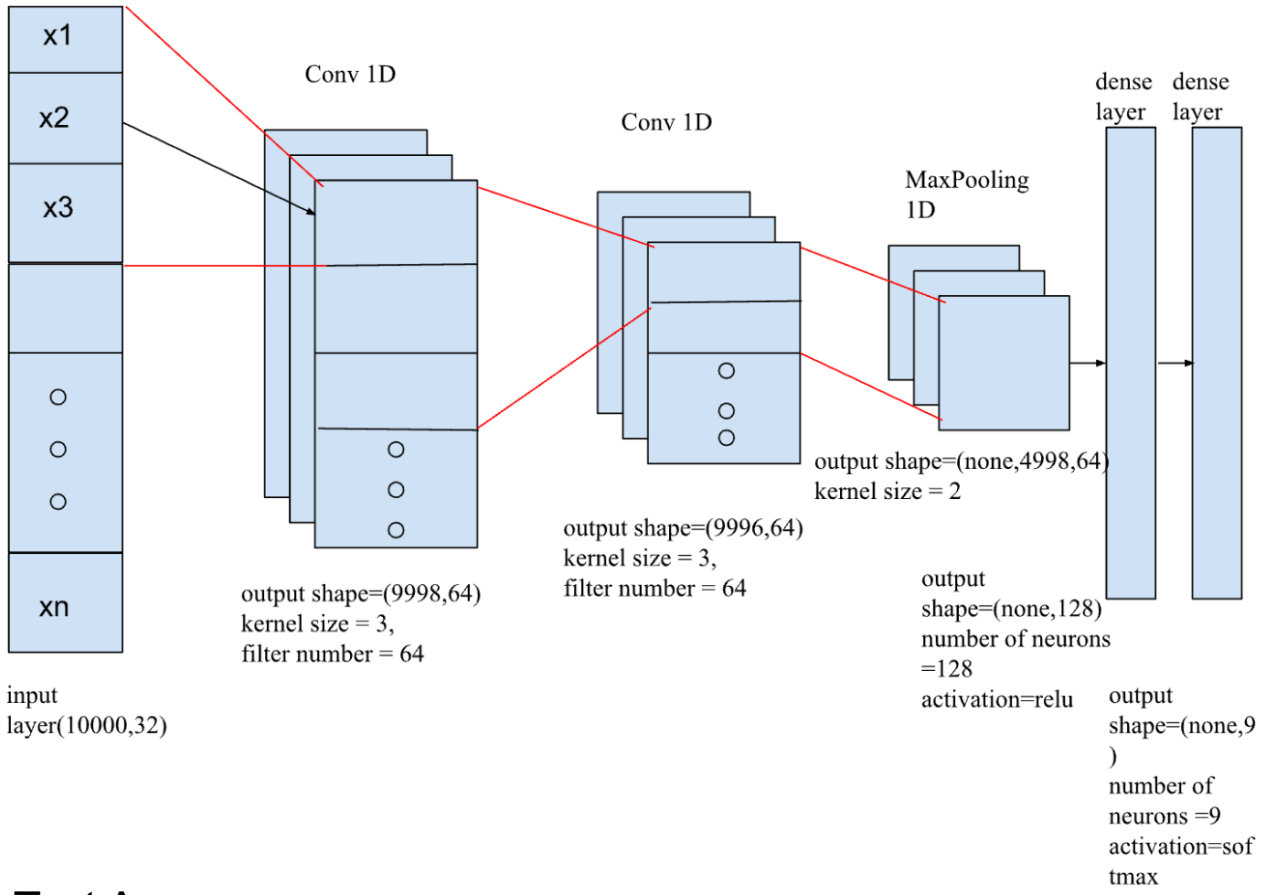
# Results: Image Classification Modality-CNN A - Confusion Matrix




| Class | Precision | Recall | F1 |
|---|---|---|---|
| gatak | 0,6060 | 0,5882 | 0,5970 |
| kelihosver 1 | 1 | 0,9428 | 0,9706 |
| kelihosver 3 | 1 | 1 | 1 |
| lollipop | 0,8214 | 0,7419 | 0,7796 |
| obfuscator | 0,8101 | 0,9697 | 0,8827 |
| ramnit | 0,9500 | 0,7716 | 0,8516 |
| simda | 0,2500 | 1 | 0,4 |
| tracur | 0,0937 | 1 | 0,1713 |
| vundo | 0,8591 | 0,8592 | 0,8591 |

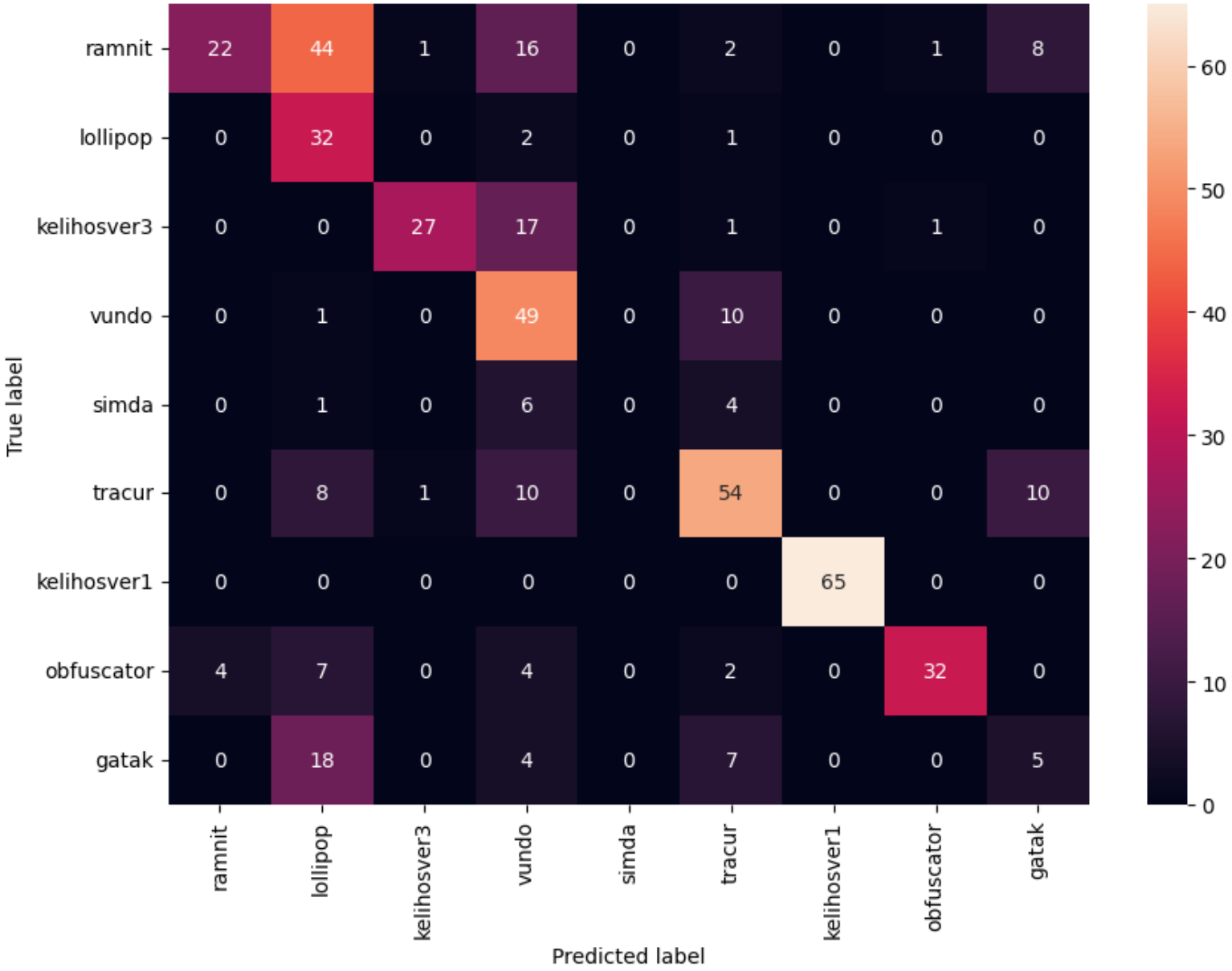
# Results: NLP Modality - CNN B



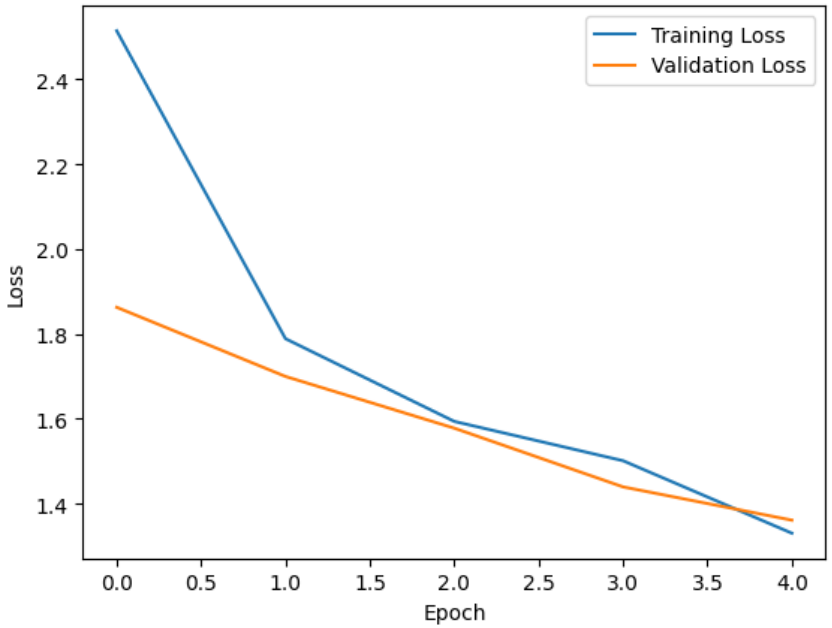

Test Accuracy:
0.6771

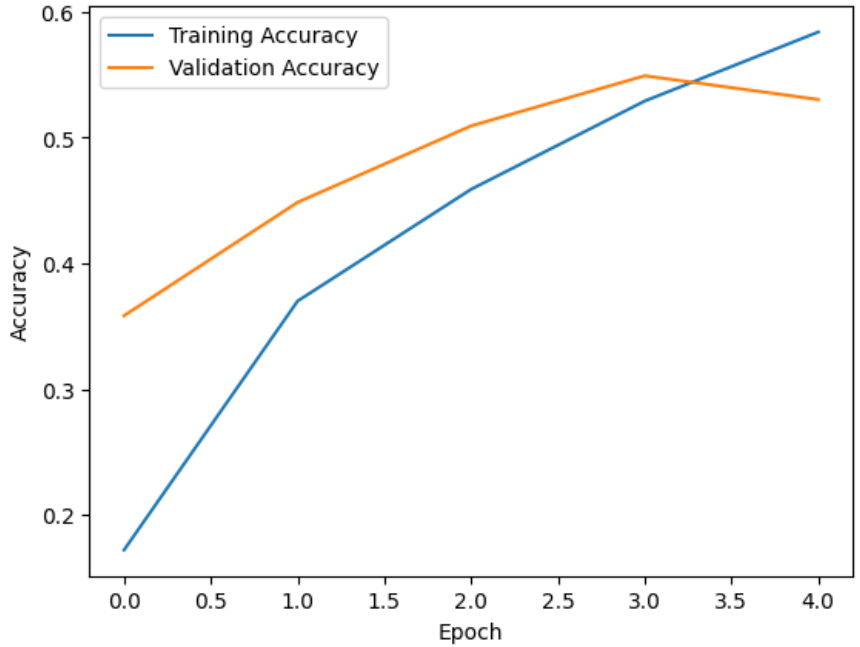
# Results: NLP Modality - CNN B

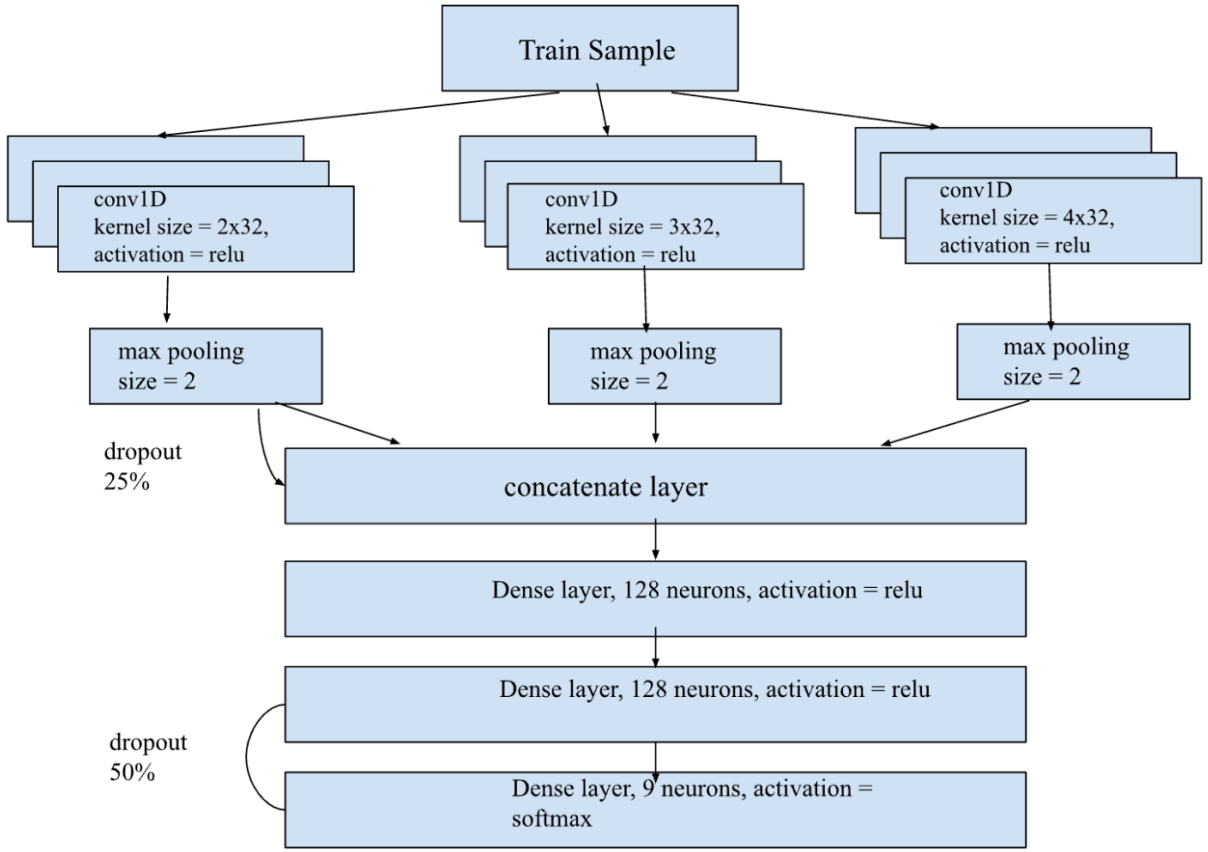



| class | precision | recall | F1 |
|---|---|---|---|
| ramnit | 0,2340 | 0,8461 | 0,3666 |
| lollipop | 0,9142 | 0,2883 | 0,4384 |
| kelihosver 3 | 0,5869 | 0,9310 | 0,7199 |
| vundo | 0,8167 | 0,4537 | 0,5833 |
| simda | 0 | undefined | undefined |
| tracur | 0,6506 | 0,6667 | 0,6585 |
| kelihosver 1 | 1 | 1 | 1 |
| obfuscator | 0,6531 | 0,9412 | 0,7711 |
| gatak | 0,1470 | 0,2174 | 0,1754 |

# Results: NLP Modality - CNN C

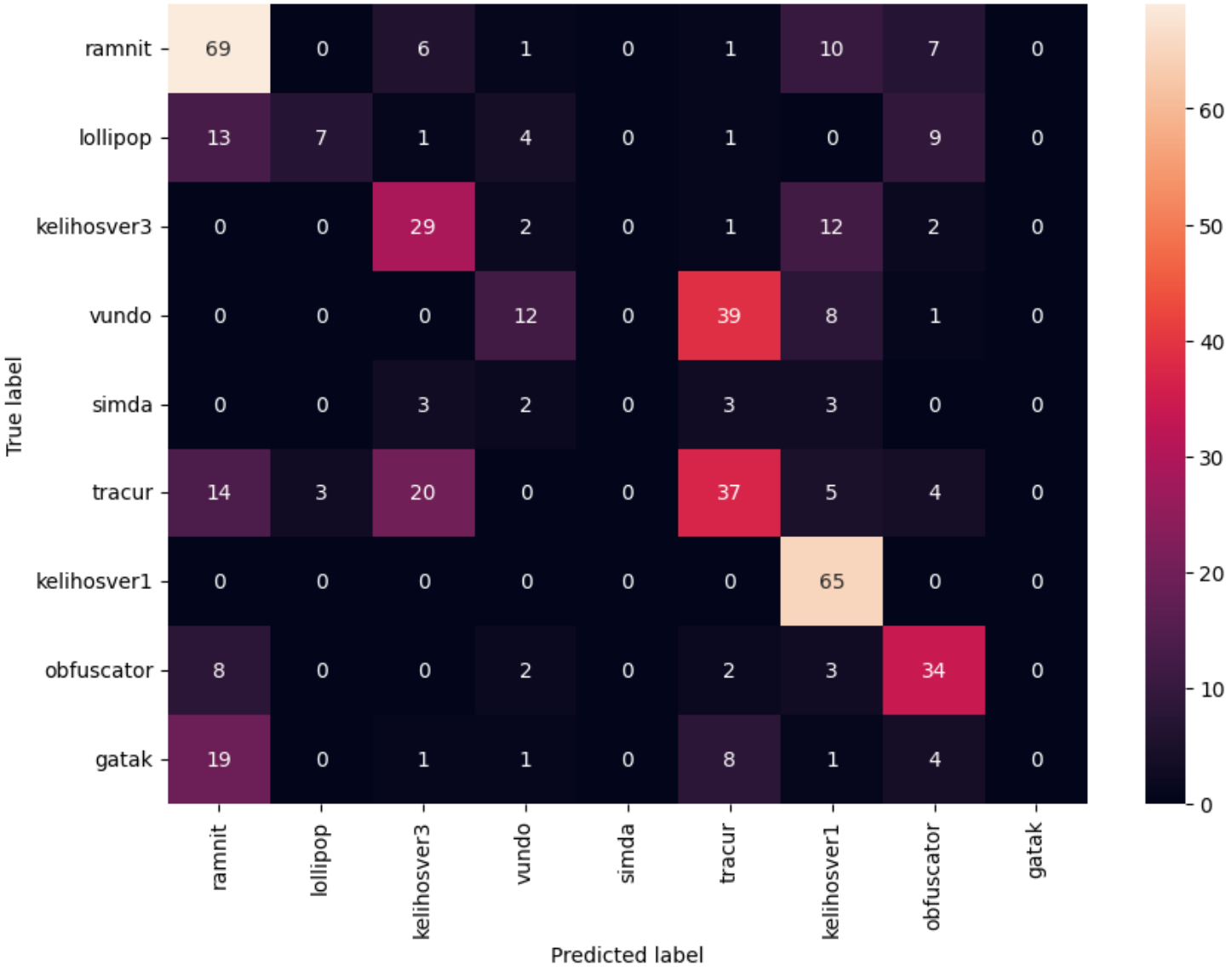


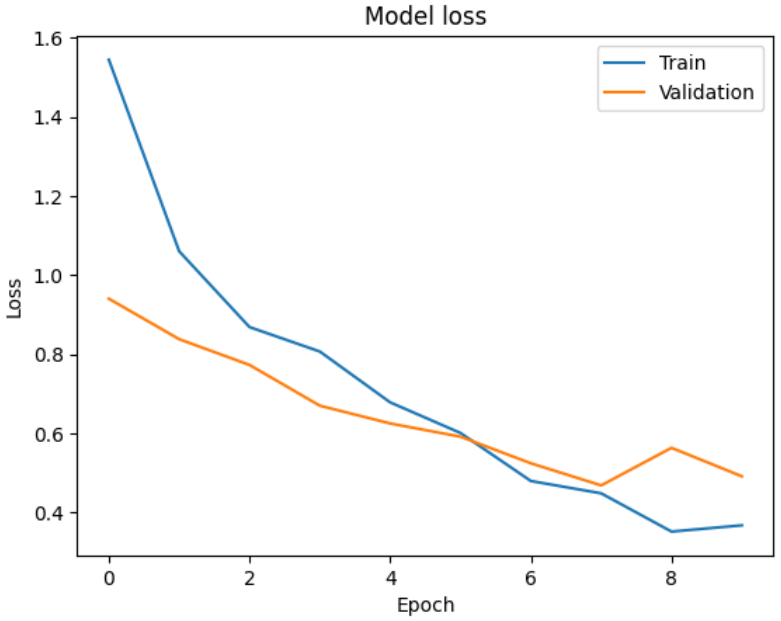

Test Accuracy:
0.5801

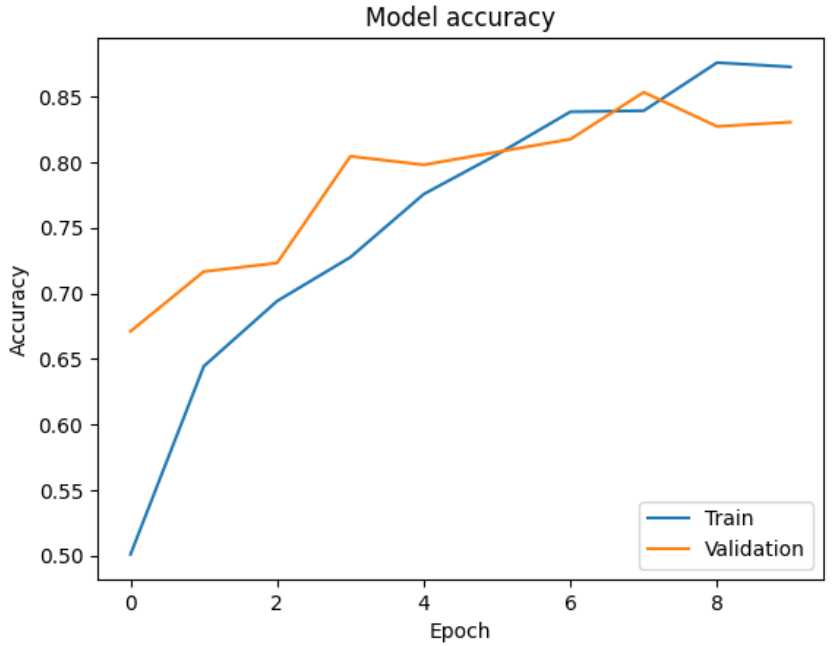
# Results: NLP Modality - CNN C

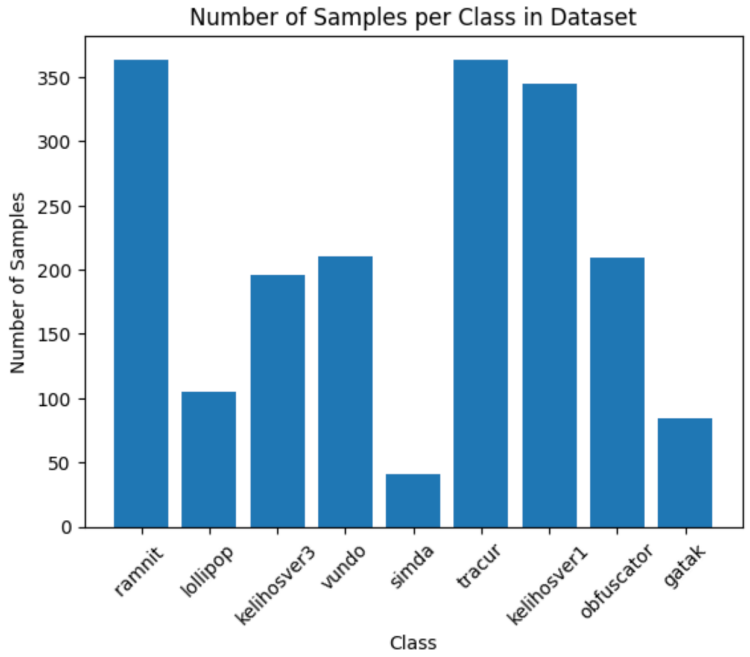


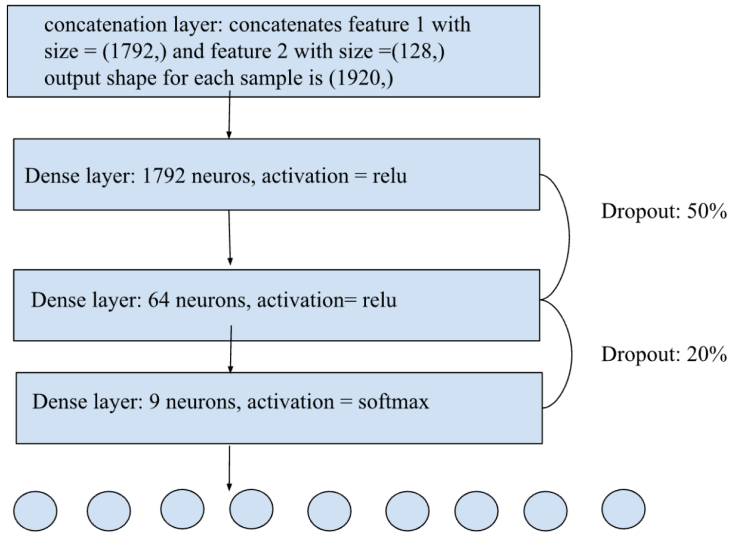

| class | precision | recall | F1 |
|---|---|---|---|
| ramnit | 0,7340 | 0,5609 | 0,6359 |
| lollipop | 0,2 | 0,7 | 0,3111 |
| kelihosver 3 | 0,6304 | 0,5088 | 0,56311 |
| vundo | 0,2 | 0,5 | 0,2857 |
| simda | 0 | undefined | undefined |
| tracur | 0,4458 | 0,4022 | 0,4229 |
| kelihosver 1 | 1 | 0,6075 | 0,7558 |
| obfuscator | 0,6939 | 0,5574 | 0,6182 |
| gatak | 0 | undefined | undefined |

# Results: Bimodal Sequential Model



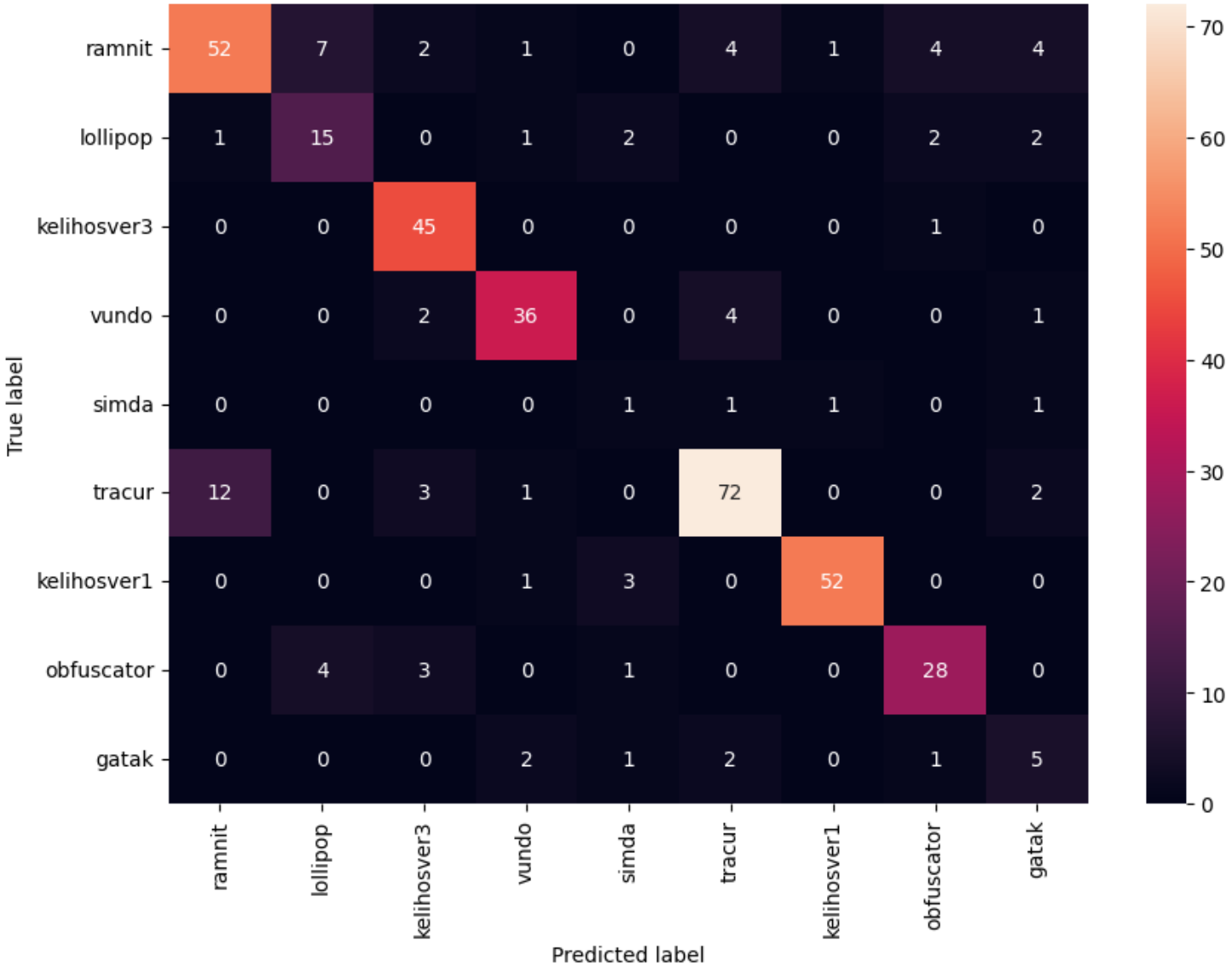




Test Accuracy: 0.8788

loss = categorical_crossentropy, optimizer = adam

# Results: Bimodal Sequential Model




| class | precision | recall | F1 |
|---|---|---|---|
| ramnit | 0,6933 | 0,8 | 0,7428 |
| lollipop | 0,6522 | 0,5769 | 0,6122 |
| kelihosver3 | 0,9782 | 0,8182 | 0,8911 |
| vundo | 0,8372 | 0,8571 | 0,8470 |
| simda | 0,25 | 0,125 | 0,1667 |
| tracur | 0,8 | 0,8675 | 0,8324 |
| kelihosver1 | 0,9286 | 0,9630 | 0,9455 |
| obfuscator | 0,7778 | 0,7778 | 0,7778 |
| gatak | 0,4545 | 0,3333 | 0,3845 |

# Conclusion

# Conclusion

- Expanding the scope of the project could involve incorporating detection and classification capabilities into the model.
- Dynamic feature extraction or extracting other features from the asm files such as function calls.
- Expand the allowable input opcodes to include the maximum number found in any file within the dataset,
- Grid search to optimize hyperparameters, identifying the best parameters for the model
- Use of class weights
- Replace CNN C with a more effective feature extraction model

# References


[1]. D. Gibert, “Convolutional Neural Networks for Malware Classification.”

[2]. M. Mimura and R. Ito, “Applying NLP techniques to malware detection in a practical environment - International Journal of Information Security,” *SpringerLink*, 06-Jun-2021. [Online]. Available: https://link.springer.com/article/10.1007/s10207-021-00553-8.

[3]. M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang and F. Iqbal, "Malware Classification with Deep Convolutional Neural Networks," 2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 2018, pp. 1-5, doi: 10.1109/NTMS.2018.8328749.

[4]. D. Gibert, C. Mateu and J. Planes, "Orthrus: A Bimodal Learning Architecture for Malware Classification," 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, 2020, pp. 1-8, doi: 10.1109/IJCNN48605.2020.9206671.

Thank you very much for listening.