Detecting Brain tumors in Magnetic Resonance Imaging (MRI)

GWU

Machine Learning
Group 5

Maliha Aziz, Zongzhu Li

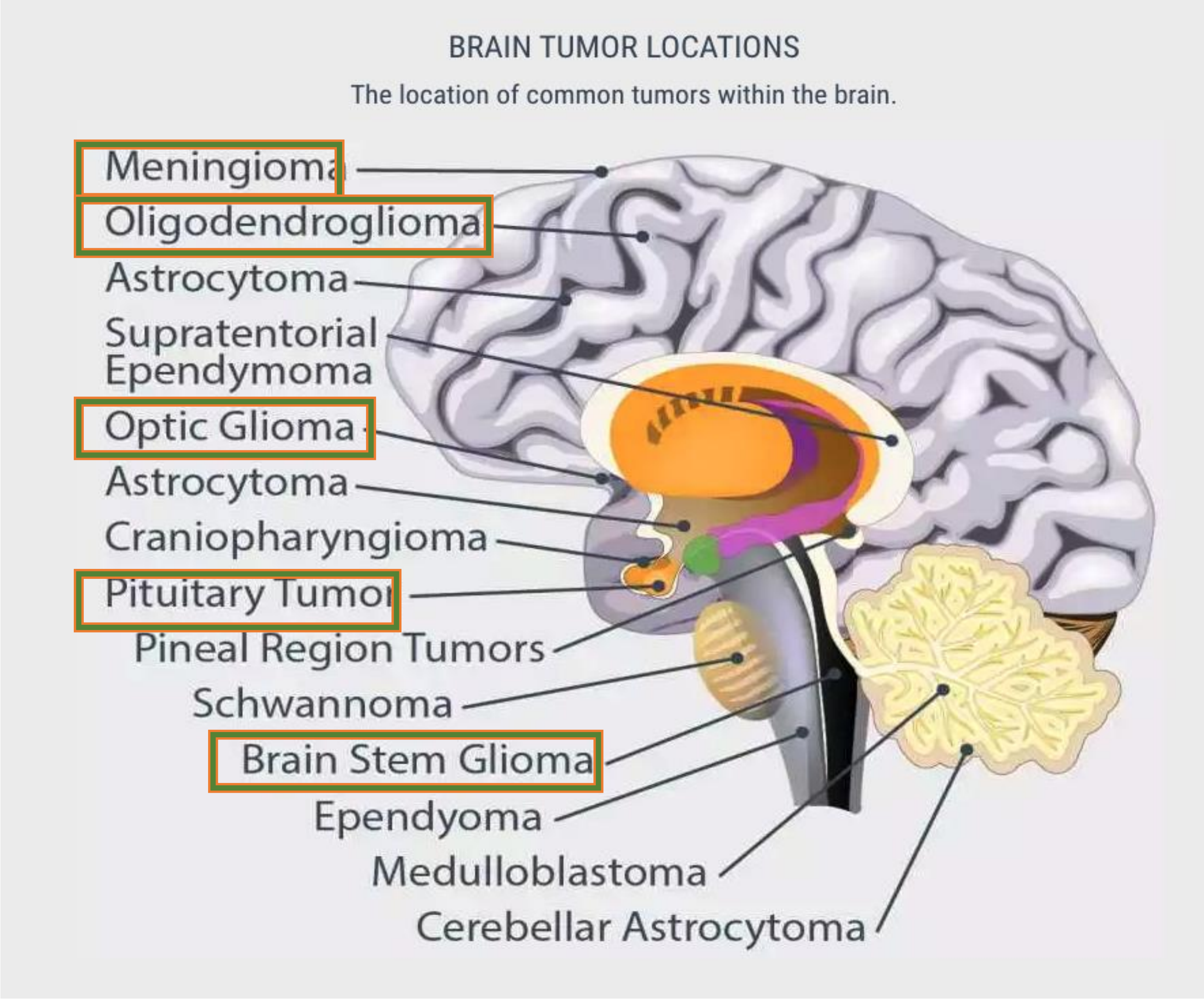
Outline

- ➢ Introduction
- ➢ Overview of Dataset for Different Types of Brain Tumors
- ➢ Data Preprocessing
- ➢ Modeling, Prediction and Evaluation
  - ➢ SVM, KNN and Naïve Bayes
  - ➢ ANN with Parameters Exploratory
- ➢ Summary and Discussion

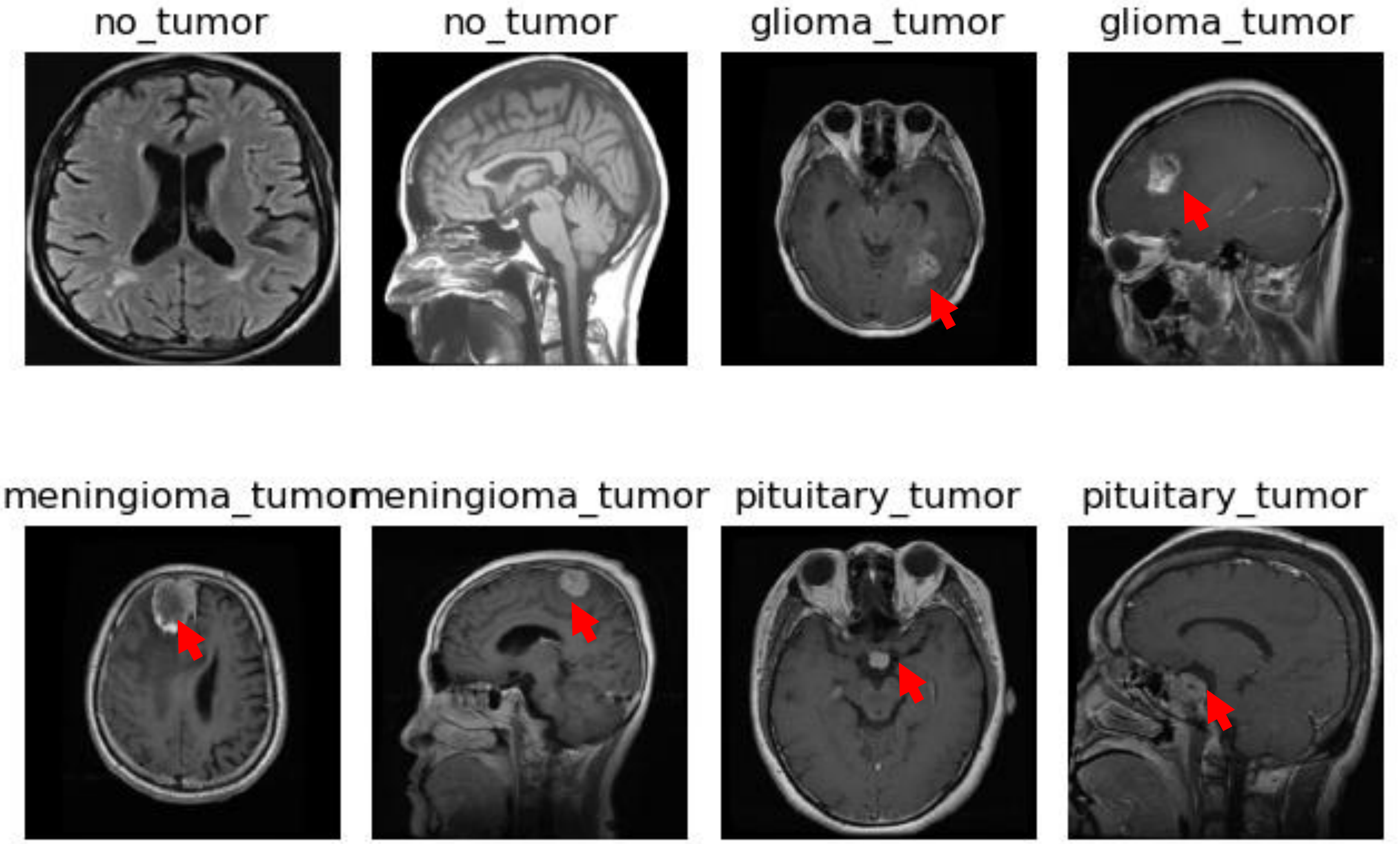
Introduction

- Challenge in Brain Cancer Diagnosis
  - Breaking through the blood-brain barrier
  - The challenges of surgical removal
  - The efficiency and accuracy of brain cancer diagnosis by doctors
- Application of Machine Learning in Brain Cancer Diagnosis
  - Assist a neuro-oncologist in diagnosing brain tumors
  - Increase the efficiency and accuracy of brain cancer diagnosis

Brain Cancer Types



Overview of Dataset for 3 Types of Brain Tumors



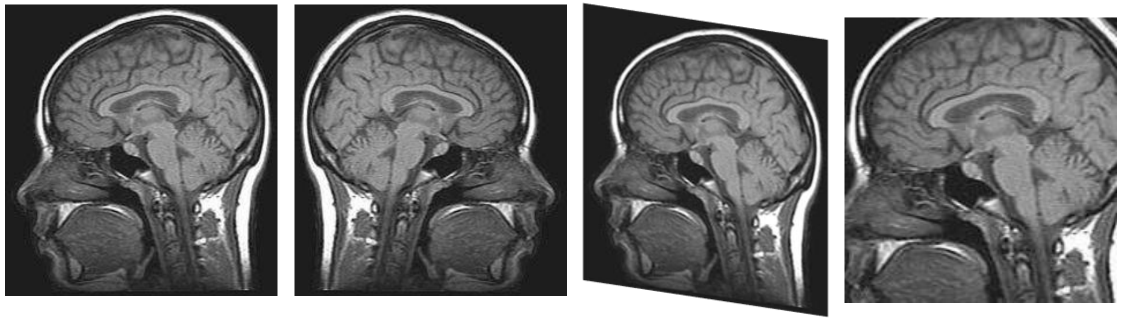
{'glioma_tumor': 826,
'meningioma_tumor': 822,
'no_tumor': 395,
'pituitary_tumor': 827}

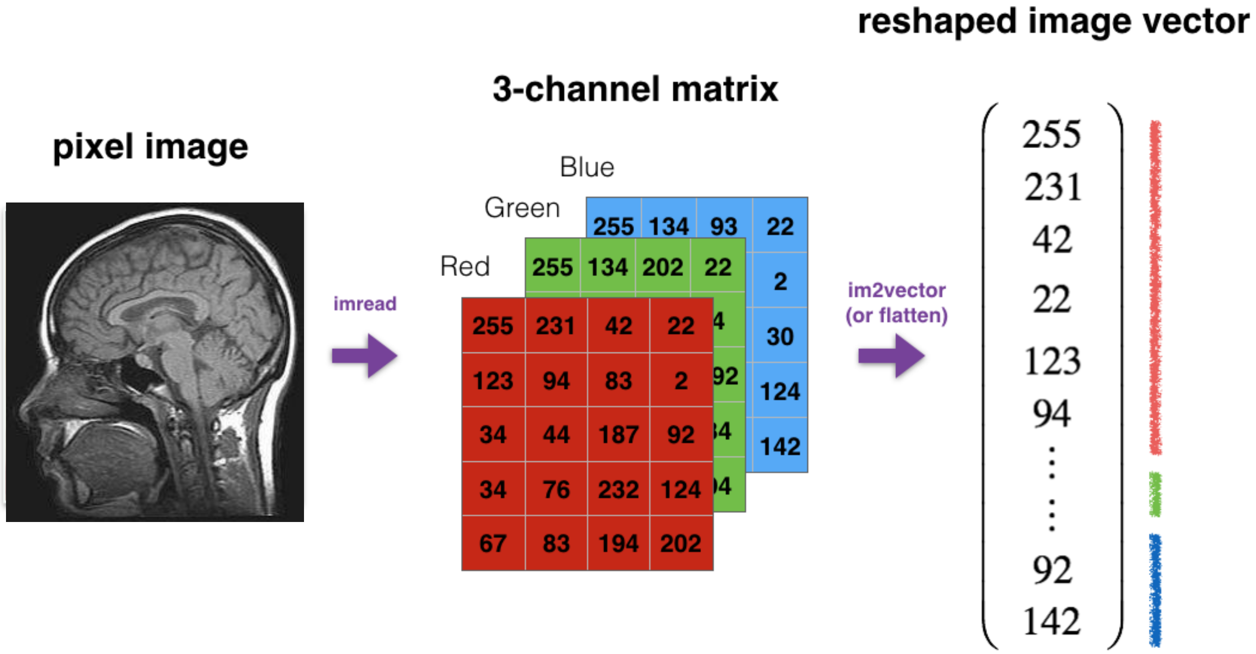
512 X 512 JPG file

https://www.kaggle.com/sartajbhuvaji/brain-tumor-classification-mri

Data Preprocessing

- Resize --- (128 X 128)
- Transform to array from image
- Normalization---(/255.0)
- Data Augmentation---(ImageDataGenerator)
  - rotation_range=20,
  - width_shift_range=0.2,
  - height_shift_range=0.2,
  - zoom_range=0.2,
  - horizontal_flip=True
- Data reshape from 3D to 2D

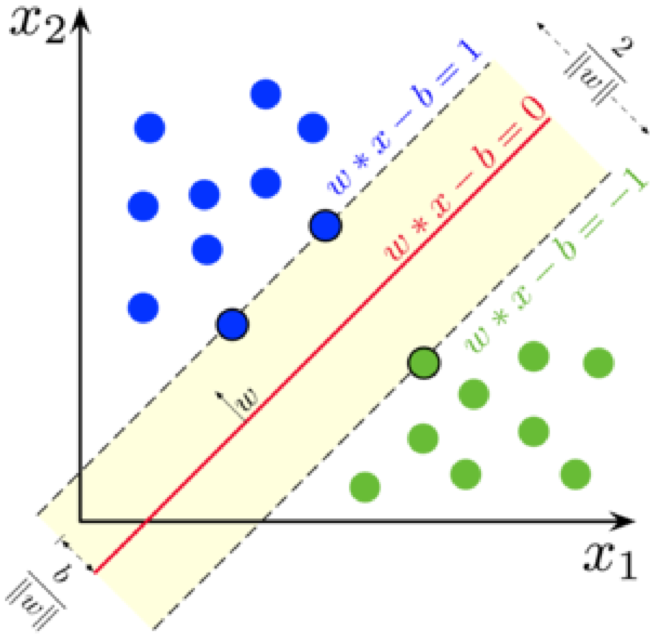


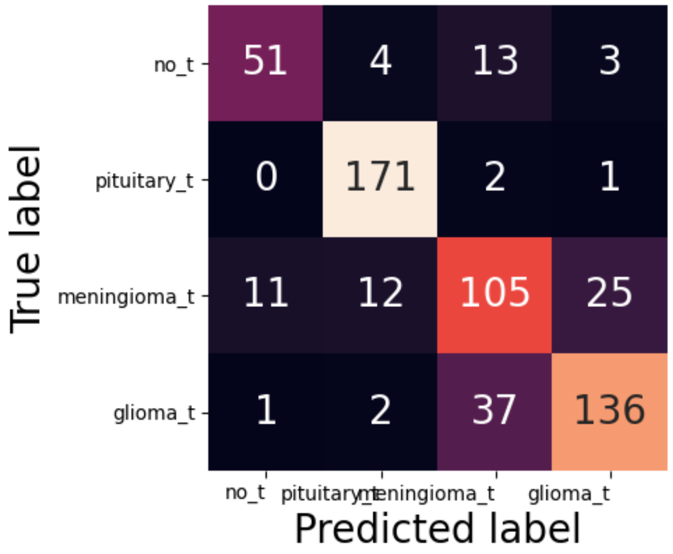


Modeling, Prediction and Evaluation

Modeling, Prediction and Evaluation -- SVM

```
clf = SVC(kernel="poly")
```

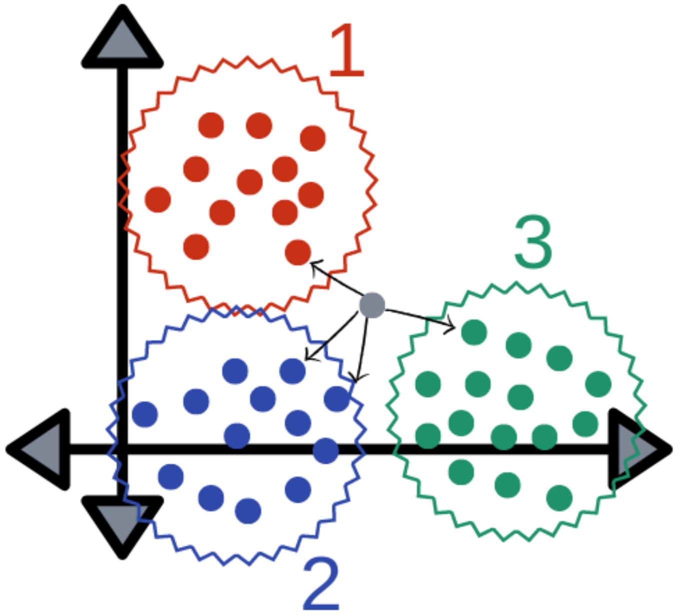


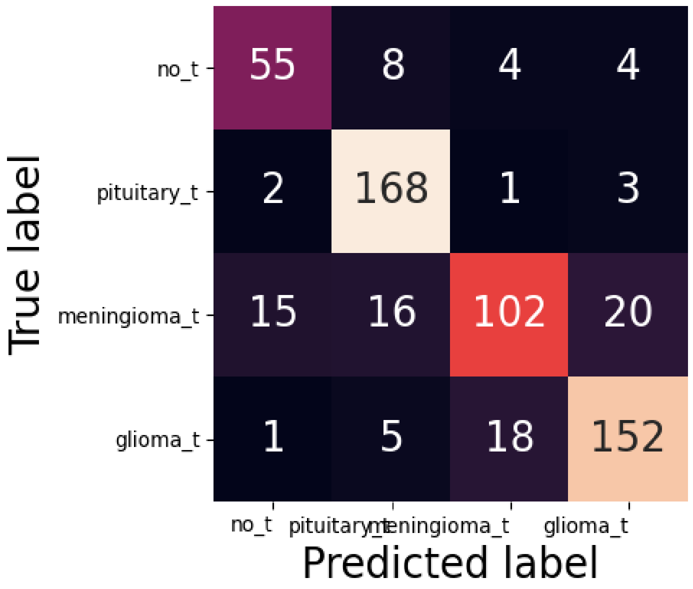



```
Classification Report:
              precision    recall  f1-score   support

           0       0.81      0.72      0.76        71
           1       0.90      0.98      0.94       174
           2       0.67      0.69      0.68       153
           3       0.82      0.77      0.80       176

    accuracy                           0.81       574
   macro avg       0.80      0.79      0.79       574
weighted avg       0.81      0.81      0.80       574
```

Modeling, Prediction and Evaluation -- KNN

clf = KNeighborsClassifier(n_neighbors=4)





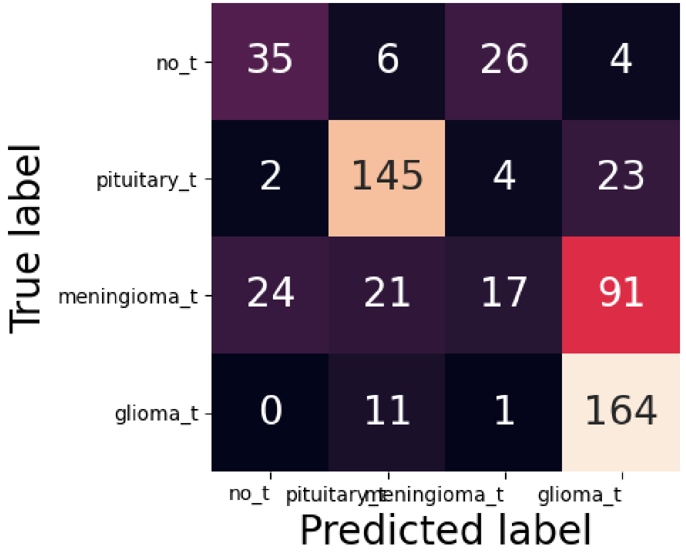
Classification Report:

	precision	recall	f1-score	support
0	0.75	0.77	0.76	71
1	0.85	0.97	0.91	174
2	0.82	0.67	0.73	153
3	0.85	0.86	0.86	176
accuracy			0.83	574
macro avg	0.82	0.82	0.81	574
weighted avg	0.83	0.83	0.83	574

Modeling, Prediction and Evaluation -- Naïve Bayes

clf = GaussianNB()

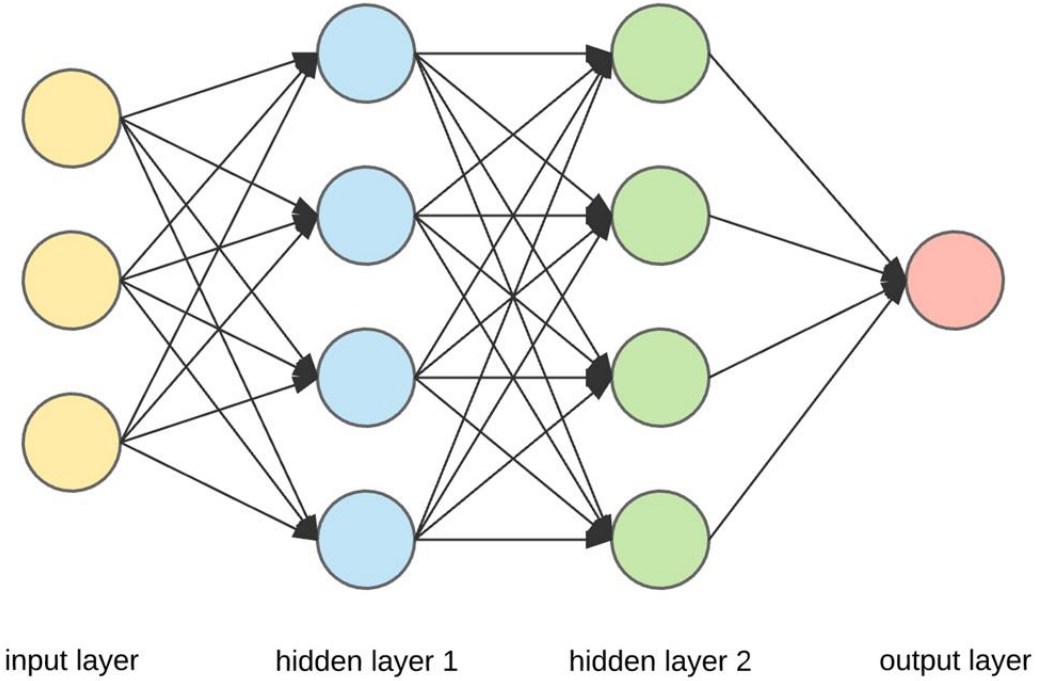
$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$



Classification Report:

	precision	recall	f1-score	support
0	0.57	0.49	0.53	71
1	0.79	0.83	0.81	174
2	0.35	0.11	0.17	153
3	0.58	0.93	0.72	176
accuracy			0.63	574
macro avg	0.58	0.59	0.56	574
weighted avg	0.58	0.63	0.58	574

Modeling, Prediction and Evaluation -- ANN



ANN optimum model discovery

1. Hidden layers and nodes

   (20,20)/ (500,500) /(20,20,20,20)

2. Iteration numbers

   500/ 2000

3. Activation

   "sigmoid"/ "relu"

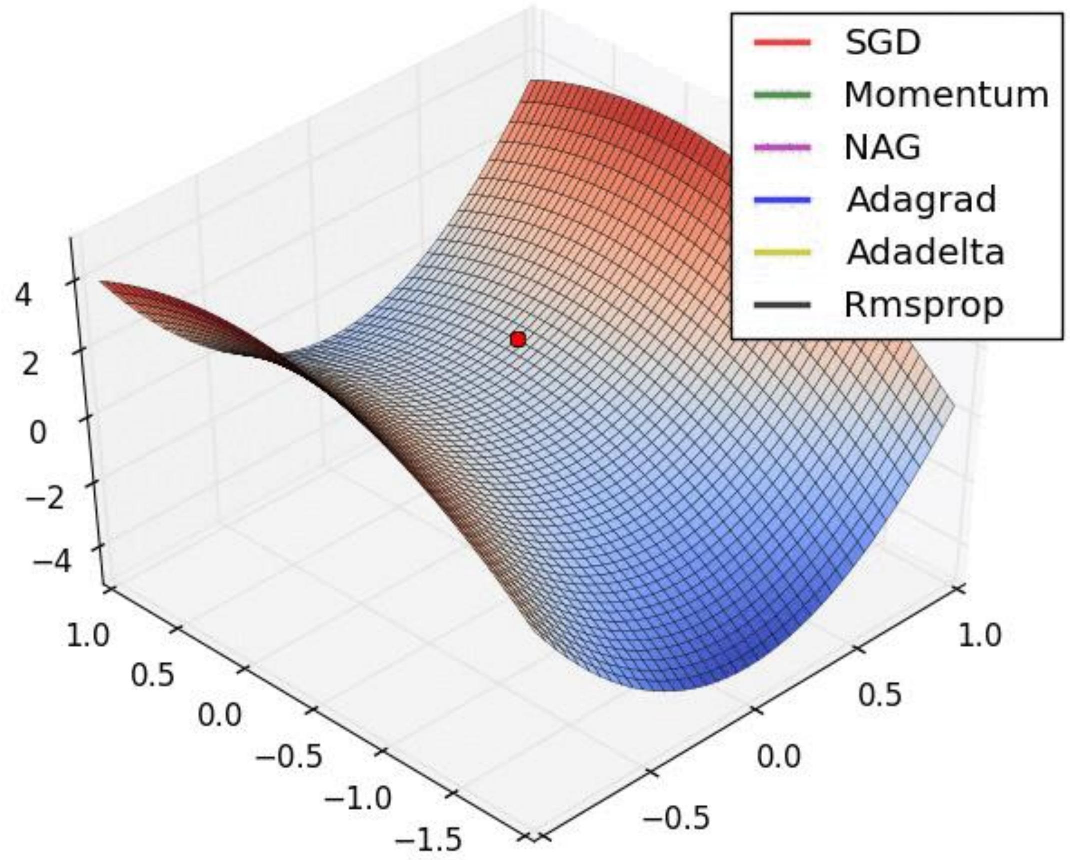
4. Early-stopping
5. Solver

   Adam/SGD/RMSprop/Adagra/Adadelta/Adamax

Optimizers

Method	Update equation
SGD	$g_t = \nabla_{\theta_t} J(\theta_t)$ $\Delta\theta_t = -\eta \cdot g_t$ $\theta_t = \theta_t + \Delta\theta_t$
Momentum	$\Delta\theta_t = -\gamma\ v_{t-1} - \eta g_t$
NAG	$\Delta\theta_t = -\gamma\ v_{t-1} - \eta \nabla_\theta J(\theta - \gamma v_{t-1})$
Adagrad	$\Delta\theta_t = -\frac{\eta}{\sqrt{G_t + \epsilon}} \odot g_t$
Adadelta	$\Delta\theta_t = -\frac{RMS[\Delta\theta]_{t-1}}{RMS[g]_t} g_t$
RMSprop	$\Delta\theta_t = -\frac{\eta}{\sqrt{E[g^2]_t + \epsilon}} g_t$
Adam	$\Delta\theta_t = -\frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$

Source: Sebastian Ruder

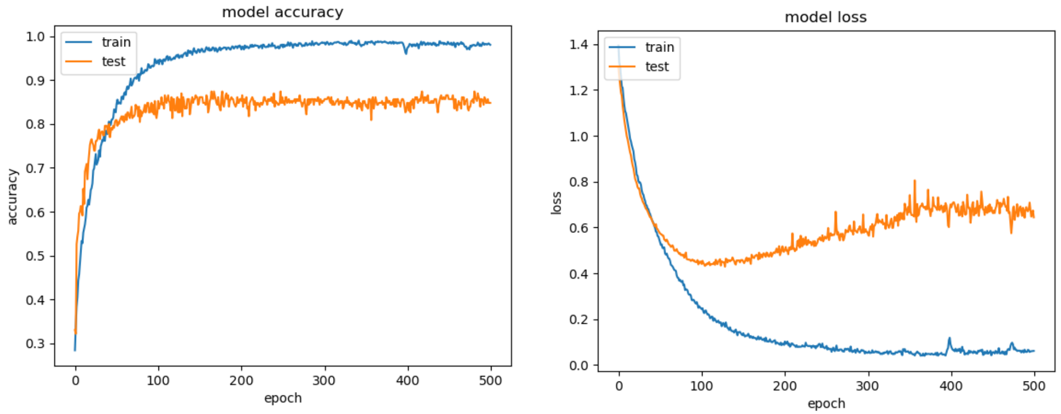


Source: Alec Radford

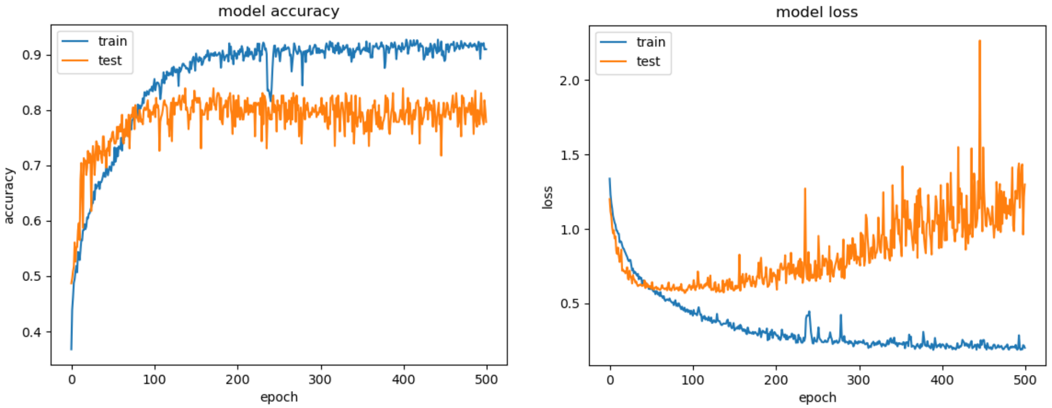
Experiments Design and Results for ANN with Different Parameters

ANN	Layers & Nodes	Iteration	Activation	Early-stopping	Solver	Accuracy (%)
01	(20, 20)	500	“sigmoid”	False	Adam	84.5
02	(20, 20)	500	“relu”	False	Adam	79.4
03	(20, 20)	500	“sigmoid”, ”relu”	False	Adam	82.1
04	(500, 500)	500	“sigmoid”	True	Adam	85.2
05	(20, 20, 20,20)	500	“sigmoid”	False	Adam	80.8
06	(20, 20)	2000	“sigmoid”	False	Adam	82.2
07	(20,20)	500	“sigmoid”	False	SGD	81.2
08	(20, 20)	500	“sigmoid”	False	RMSprop	84.8
09	(20, 20)	500	“sigmoid”	False	Adagrad	81.0
10	(20, 20)	500	“sigmoid”	False	Adadelta	78.6
11	(20,20)	500	“sigmoid”	False	Adamax	73.3
ANN	(500, 500, 20)	500	“sigmoid”	False	RMSprop	87.5

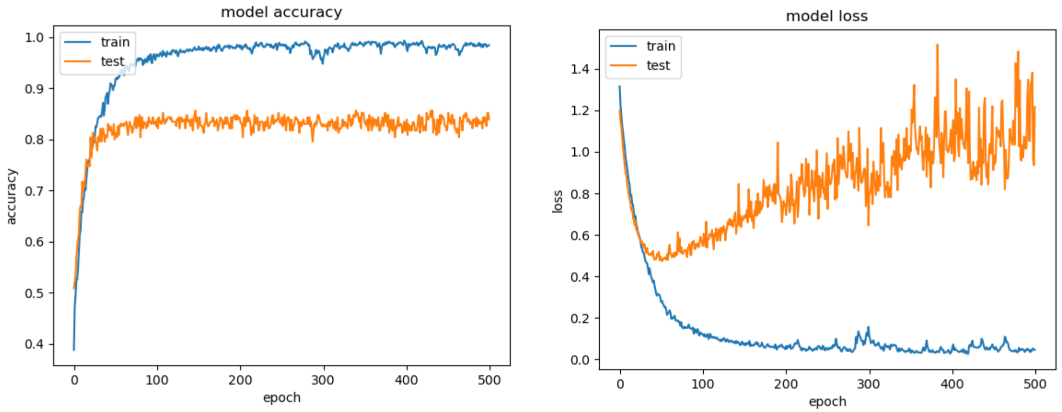
ANN 01



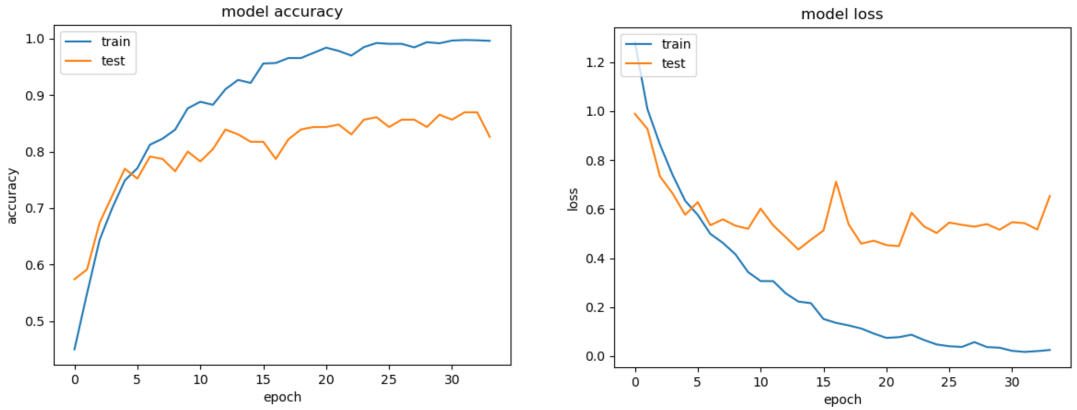
ANN 02



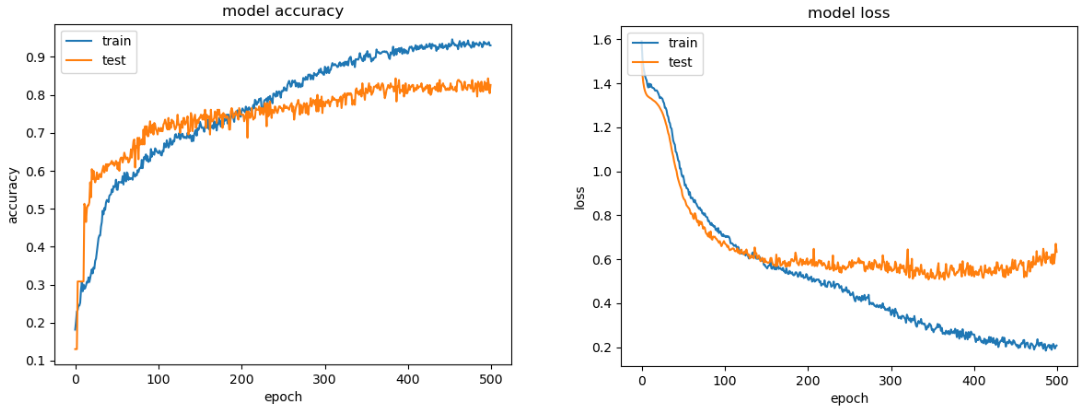
ANN 03



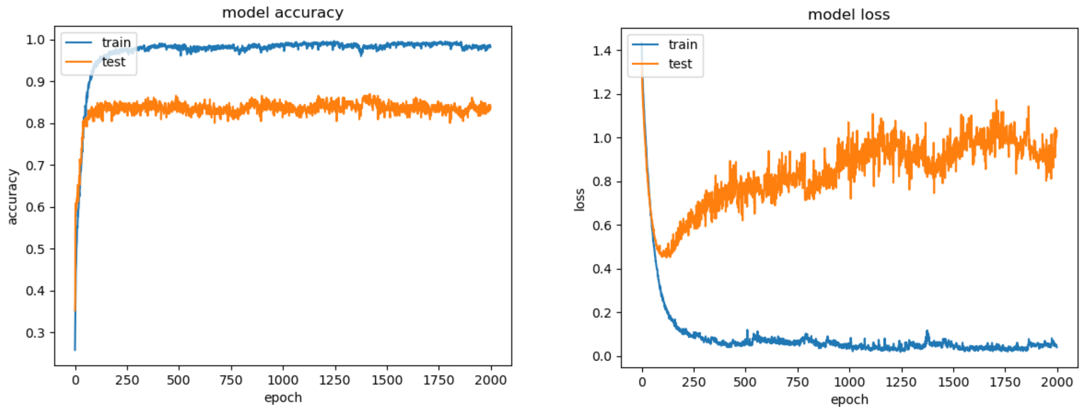
ANN 04



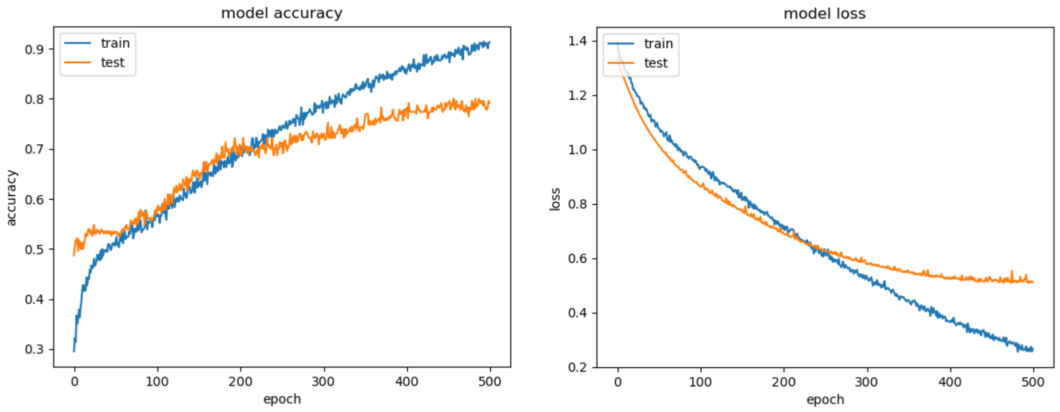
ANN 05



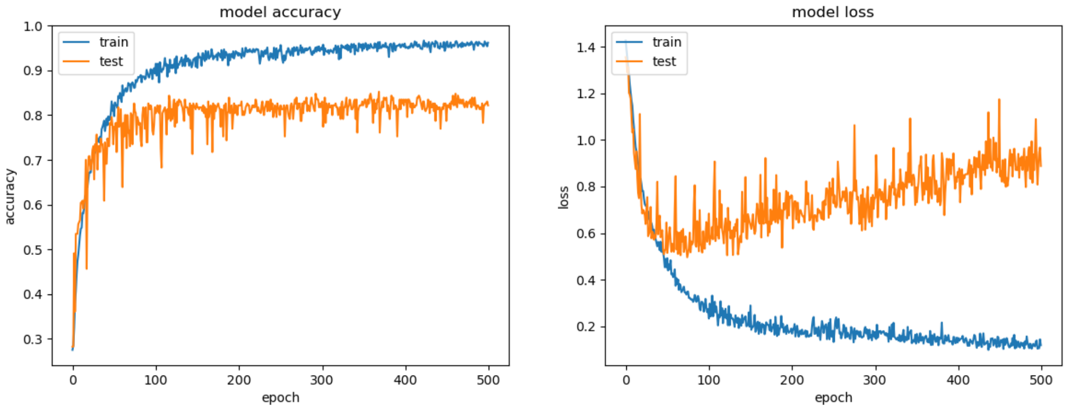
ANN 06



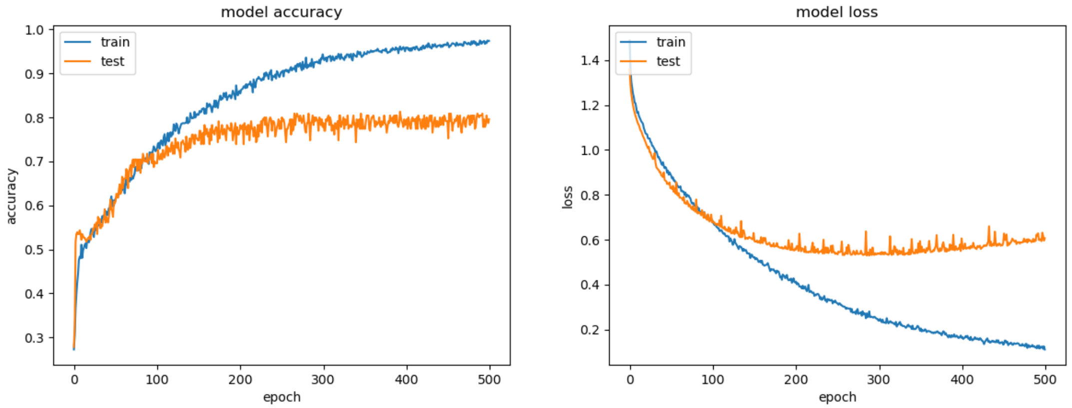
ANN 07



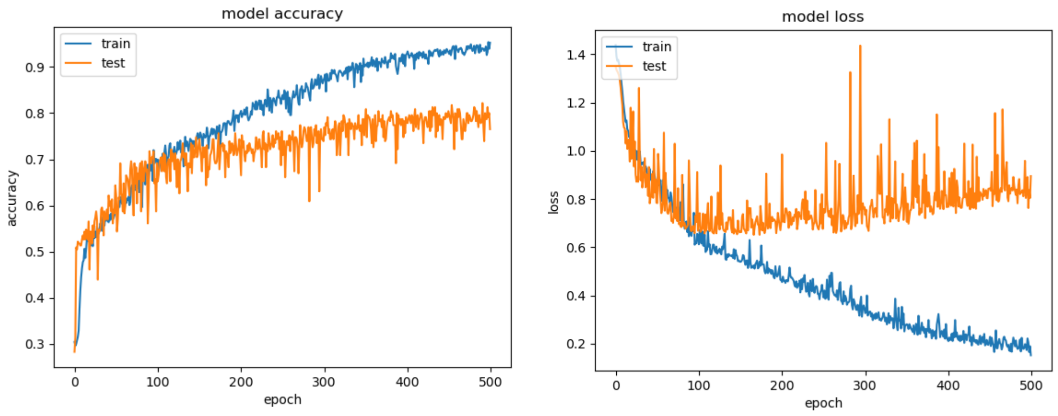
ANN 08



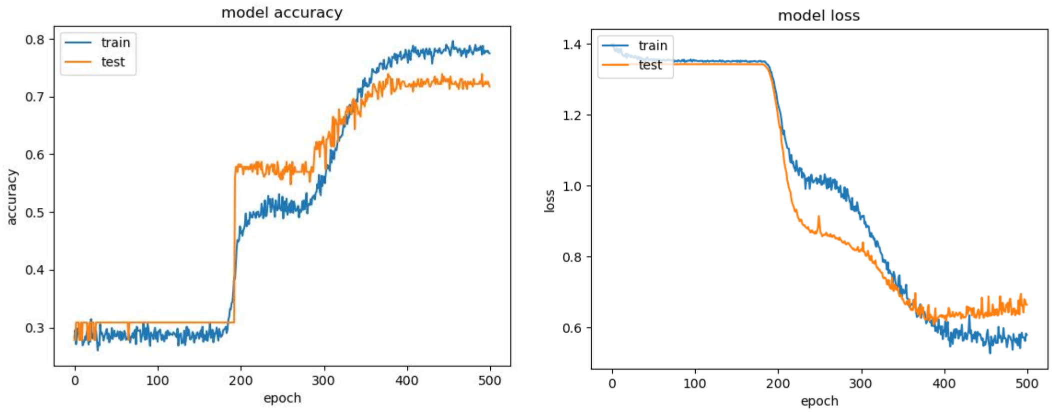
ANN 09



ANN 10

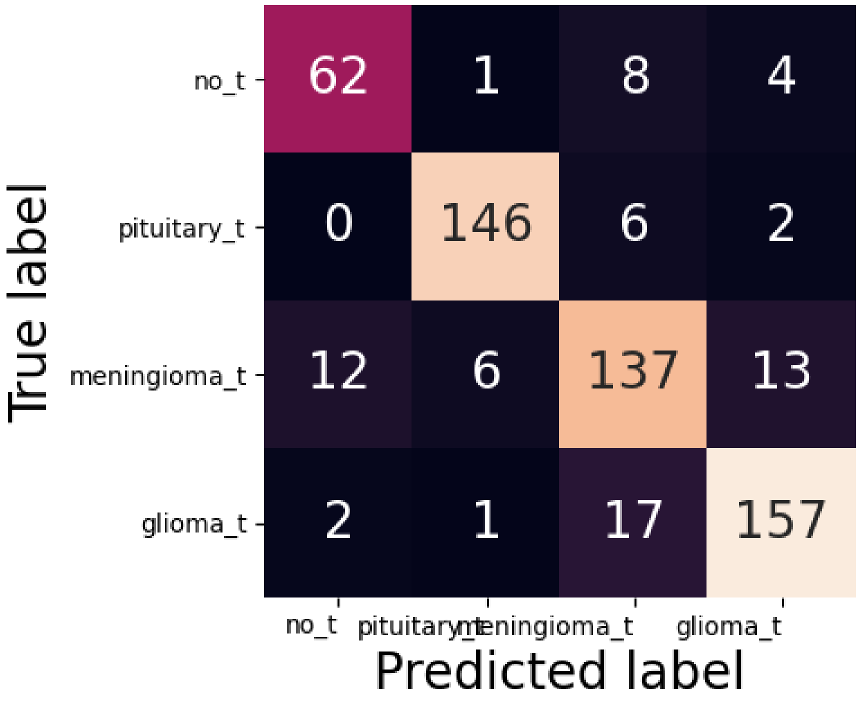


ANN 11



Modeling, Prediction and Evaluation - ANN

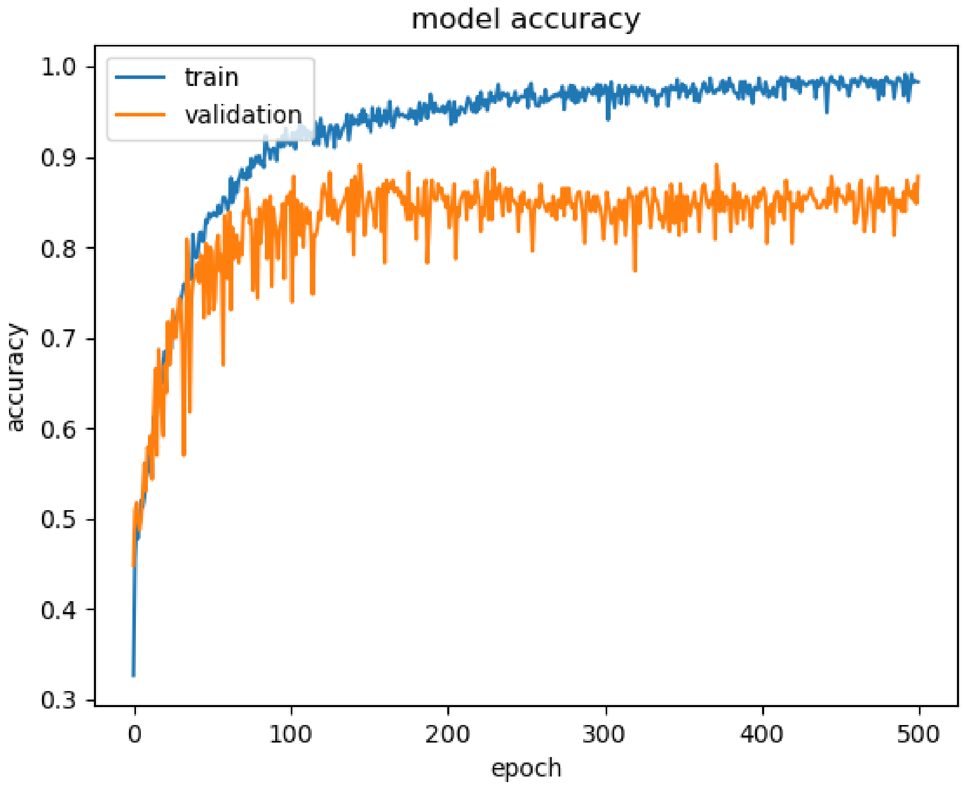
Hidden layers and Nodes: (500,500,20)/ Iteration: 500/ Activation: “sigmoid”/ Solver: RMSpro (lr=0.001)

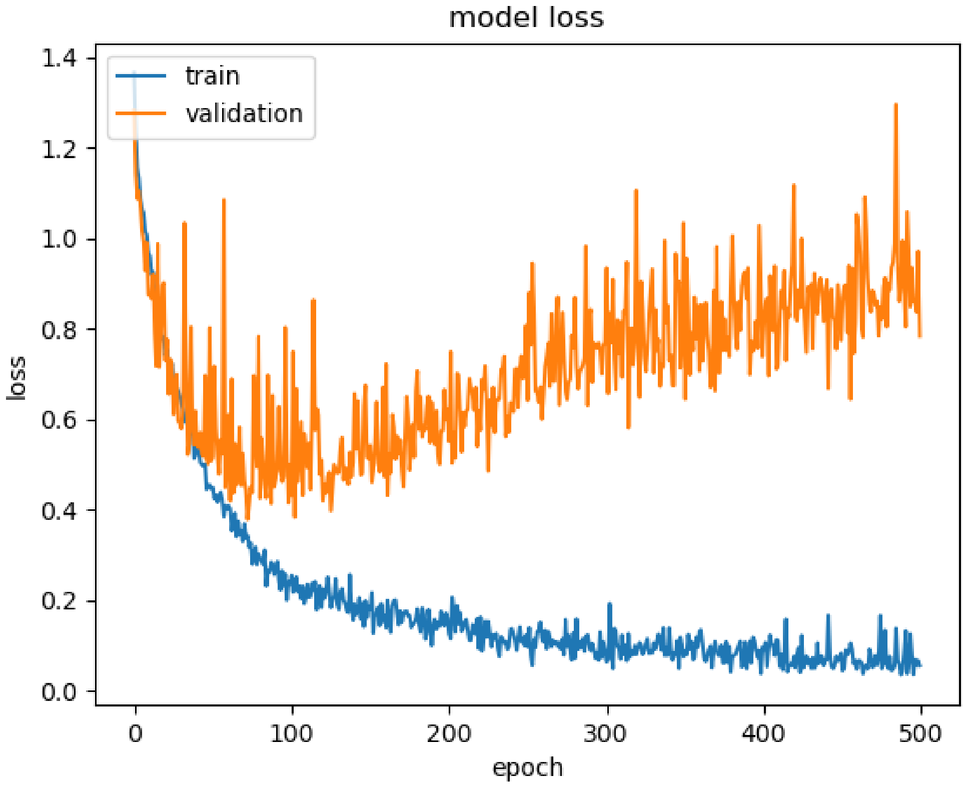


Classification Report:

	precision	recall	f1-score	support
0	0.82	0.83	0.82	75
1	0.95	0.95	0.95	154
2	0.82	0.82	0.82	168
3	0.89	0.89	0.89	177
accuracy			0.87	574
macro avg	0.87	0.87	0.87	574
weighted avg	0.87	0.87	0.87	574

Modeling, Prediction and Evaluation - ANN





Summary and Discussion

1. Activation with “sigmoid” shows better performance and higher accuracy than activation with “relu” and “tanh”.

2. Solver with RMSprop shows higher prediction accuracy with the brain cancer dataset.

3. With ANN model, the accuracy for prediction can reach to 87%, which is better than other models including KNN, SVM and Naïve Bayes.

Deep learning with skills of convolution and maxpool need to be tried to get the accuracy higher……

References

1. Aldape, K., Brindle, K.M., Chesler, L. et al. Challenges to curing primary brain tumours. Nat Rev Clin Oncol 16, 509–520 (2019). https://doi.org/10.1038/s41571-019-0177-5
2. Cao, H., Erson-Omay, E.Z., Li, X. et al. A quantitative model based on clinically relevant MRI features differentiates lower grade gliomas and glioblastoma. Eur Radiol 30, 3073–3082 (2020). https://doi.org/10.1007/s00330-019-06632-8
3. Nadeem, M. W., Ghamdi, M., Hussain, M., Khan, M. A., Khan, K. M., Almotiri, S. H., & Butt, S. A. (2020). Brain Tumor Analysis Empowered with Deep Learning: A Review, Taxonomy, and Future Challenges. Brain sciences, 10(2), 118. https://doi.org/10.3390/brainsci10020118
4. Hollon, T.C., Pandian, B., Adapa, A.R. *et al.* Near real-time intraoperative brain tumor diagnosis using stimulated Raman histology and deep neural networks. *Nat Med* **26,** 52–58 (2020). https://doi.org/10.1038/s41591-019-0715-9
5. Kingma, D.P., & Ba, J. (2015). Adam: A Method for Stochastic Optimization. CoRR, abs/1412.6980.
6. https://ml-cheatsheet.readthedocs.io/en/latest/index.html
7. https://ruder.io/optimizing-gradient-descent/index.html#adamax
8. https://miamineurosciencecenter.com/en/conditions/brain-tumors/types/
9. https://mlfromscratch.com/optimizers-explained/#/

Shameless Plug

Help us save the world (by improving public health)

GW | Student Life | University News | Research & Learning | Politics & Society | COVID-19 Research |

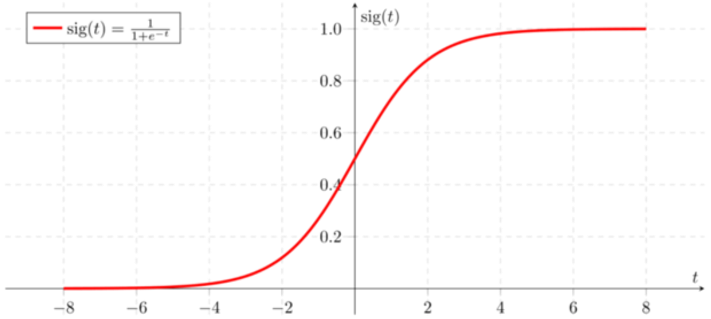
The Lab Behind GW's COVID Testing Program

A team of Milken Institute School of Public Health researchers, led by Professor Cindy Liu, ensure GW's COVID-19 testing provides rapid, consistent and accurate results.

A team of GW researchers process thousands of COVID-19 tests a week as part of a new public health laboratory, led by Associate

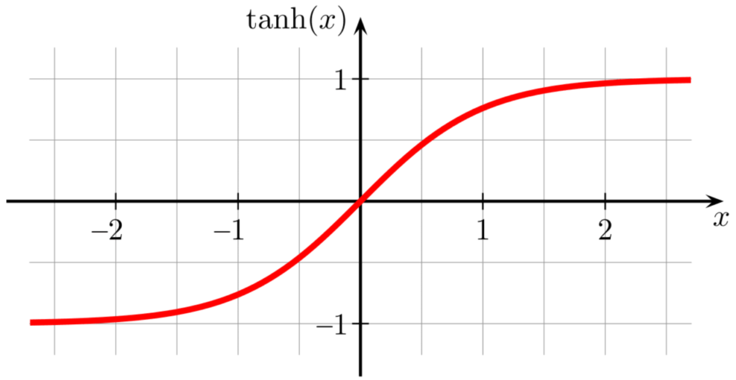
Thank You

sigmoid



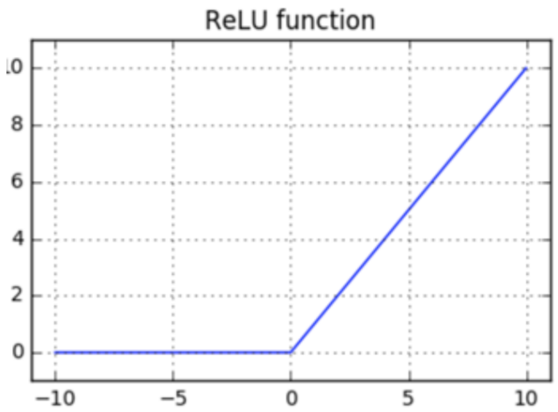
$y = 1/(1+e^{(-x)})$

tanh



$y = tanh(x)$

relu



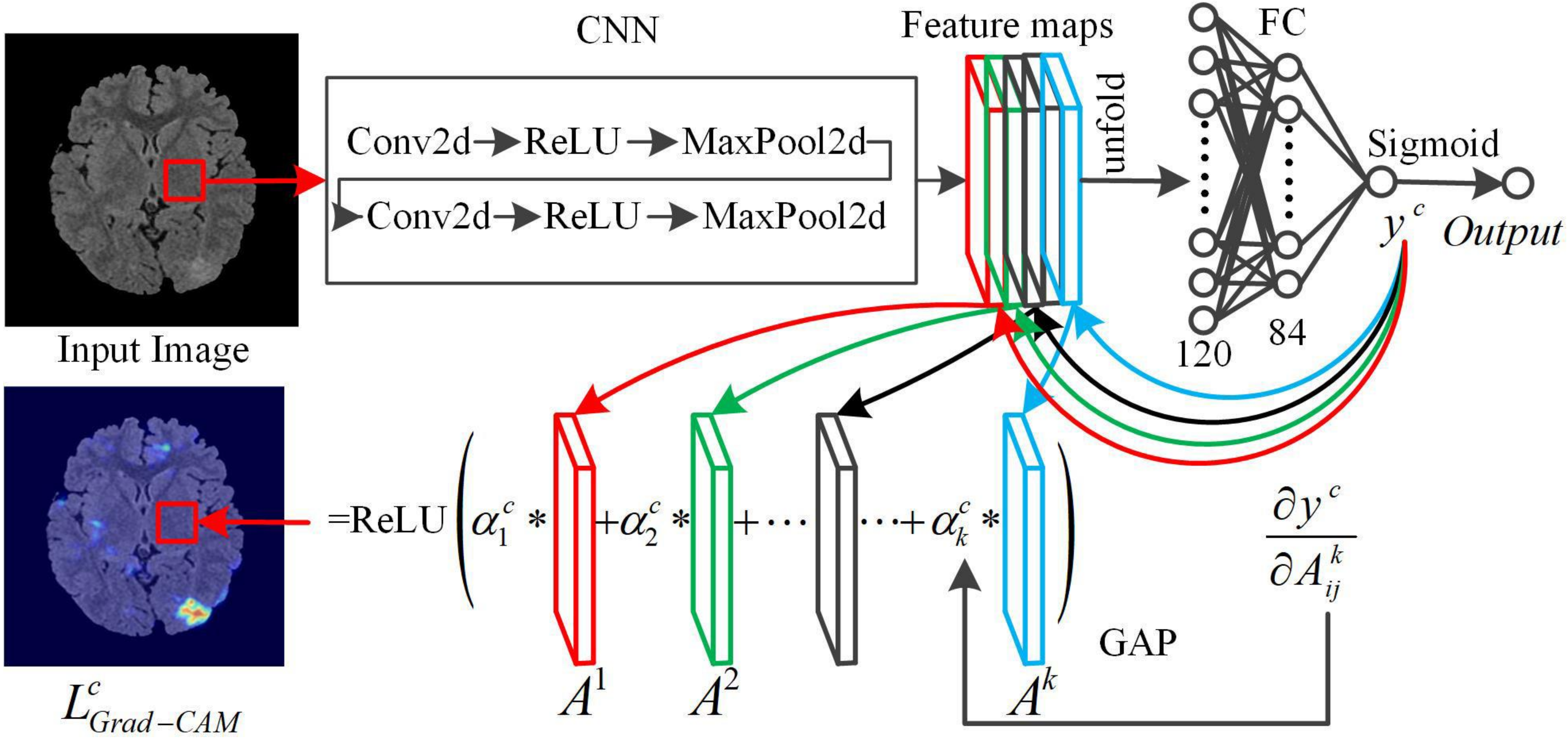
{ xi if x >=0
0 if x <=0 }

https://www.aitude.com/comparison-of-sigmoid-tanh-and-relu-activation-functions/

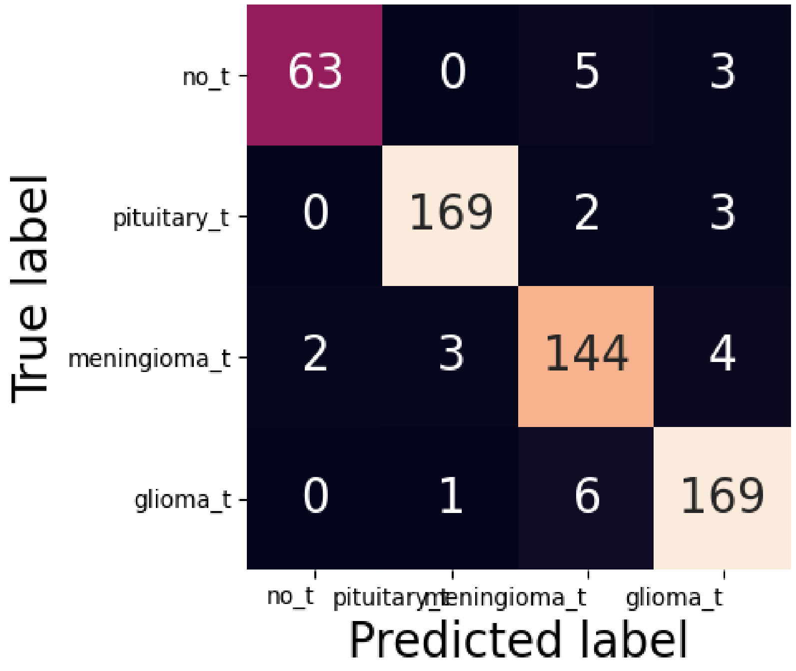
Sigmoid: not blowing up activation

Relu : not vanishing gradient

Modeling, Prediction and Evaluation - CNN



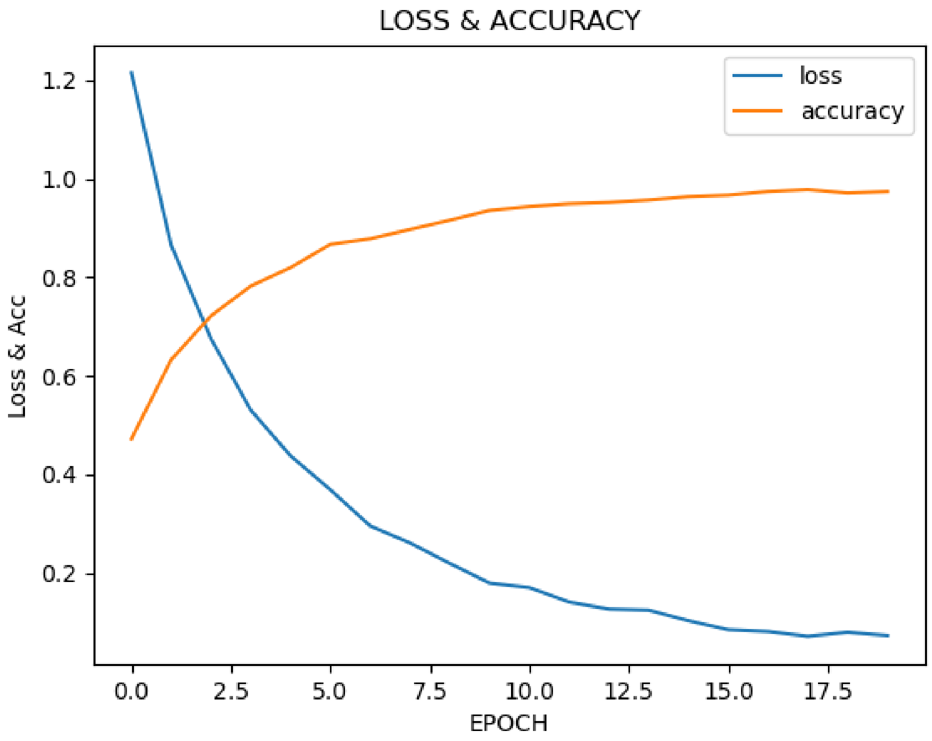
Modeling, Prediction and Evaluation - CNN



Classification Report:

	precision	recall	f1-score	support
0	0.97	0.89	0.93	71
1	0.98	0.97	0.97	174
2	0.92	0.94	0.93	153
3	0.94	0.96	0.95	176
accuracy			0.95	574
macro avg	0.95	0.94	0.95	574
weighted avg	0.95	0.95	0.95	574

Modeling, Prediction and Evaluation - CNN




```
Epoch 9/50
72/72 [==============================] - 339s 5s/step - loss: 0.2201 - accuracy: 0.9155
Epoch 10/50
72/72 [==============================] - 315s 4s/step - loss: 0.1801 - accuracy: 0.9355
Epoch 11/50
72/72 [==============================] - 310s 4s/step - loss: 0.1710 - accuracy: 0.9434
Epoch 12/50
72/72 [==============================] - 312s 4s/step - loss: 0.1416 - accuracy: 0.9490
Epoch 13/50
72/72 [==============================] - 318s 4s/step - loss: 0.1273 - accuracy: 0.9517
Epoch 14/50
72/72 [==============================] - 315s 4s/step - loss: 0.1252 - accuracy: 0.9564
Epoch 15/50
72/72 [==============================] - 306s 4s/step - loss: 0.1038 - accuracy: 0.9634
Epoch 16/50
72/72 [==============================] - 305s 4s/step - loss: 0.0855 - accuracy: 0.9665
Epoch 17/50
72/72 [==============================] - 323s 4s/step - loss: 0.0819 - accuracy: 0.9739
Epoch 18/50
72/72 [==============================] - 317s 4s/step - loss: 0.0722 - accuracy: 0.9778
Epoch 19/50
72/72 [==============================] - 455s 6s/step - loss: 0.0803 - accuracy: 0.9708
Epoch 20/50
72/72 [==============================] - 391s 5s/step - loss: 0.0737 - accuracy: 0.9739
```