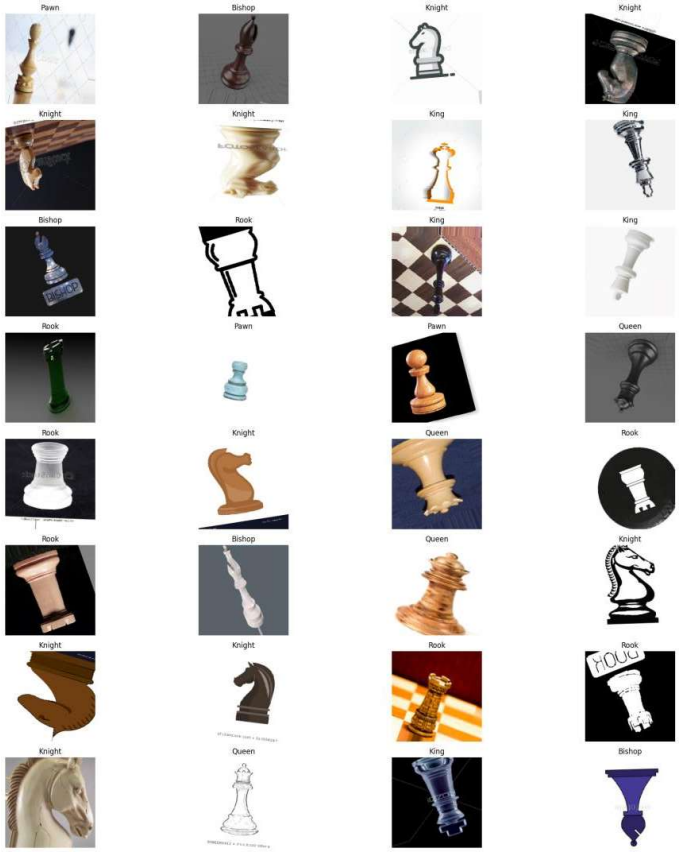
AKT – Group Project

Fabian Hirschmann, Michaela Hauer

DataSet

- Source: https://www.kaggle.com/datasets/niteshfre/chessman-image-dataset/data
- 6 Classes:
  - Bishop,
  - King,
  - Knight,
  - Pawn,
  - Queen,
  - Rook
- 556 files

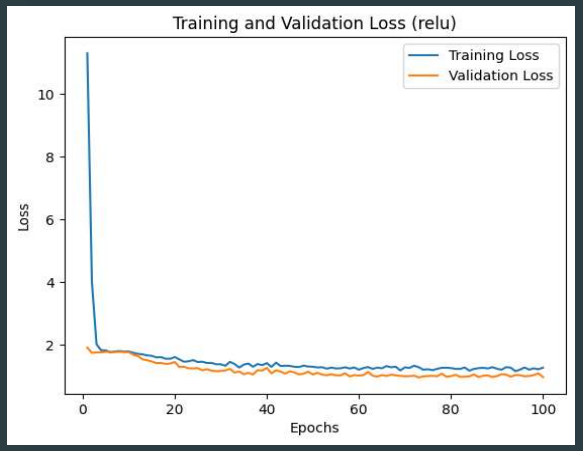


Evaluation of activation functions:

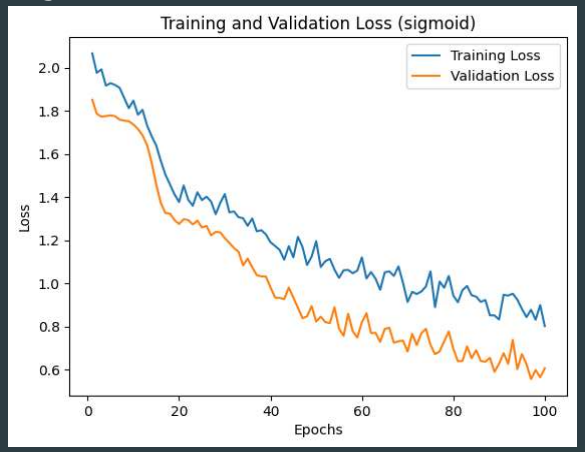
- relu
- sigmoid
- tanh
- selu
- elu
- exponential
- leaky_relu
- silu
- gelu
- hard_sigmoid
- linear
- mish

categorical_accuracy (loss in training)

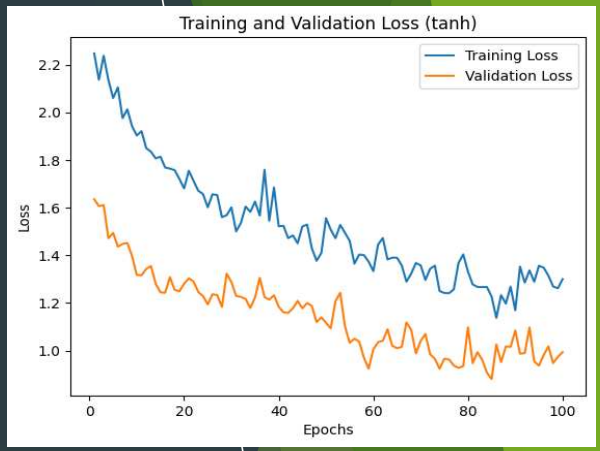
relu



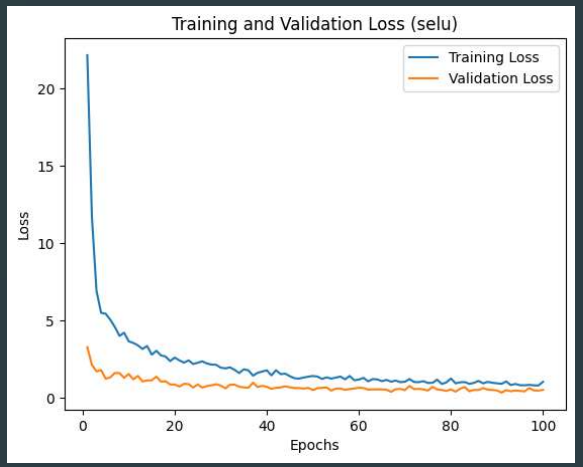
sigmoid



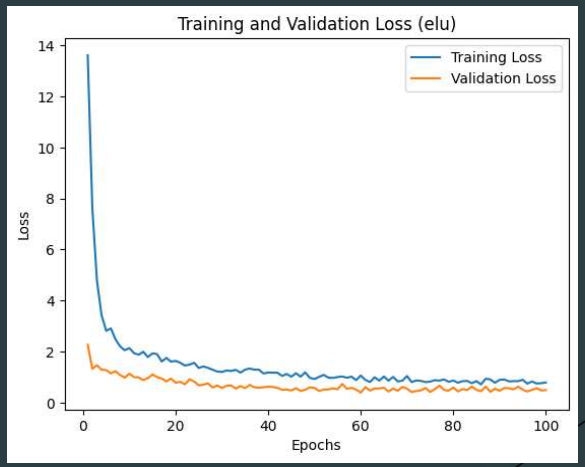
tanh



silu

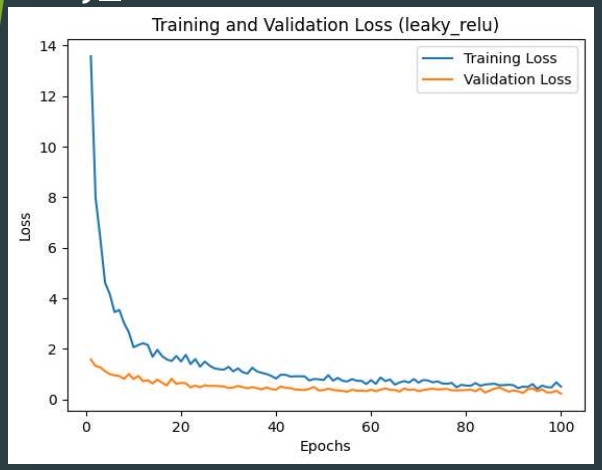


elu

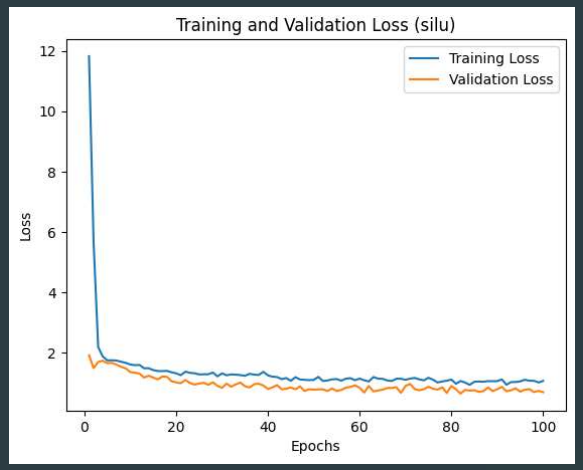


categorical_accuracy (loss in training)

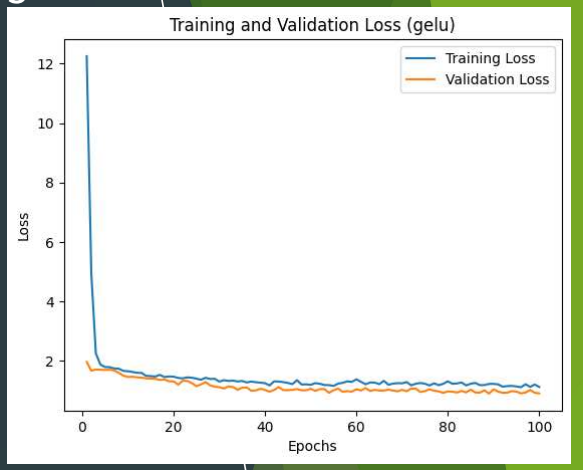
leaky_relu



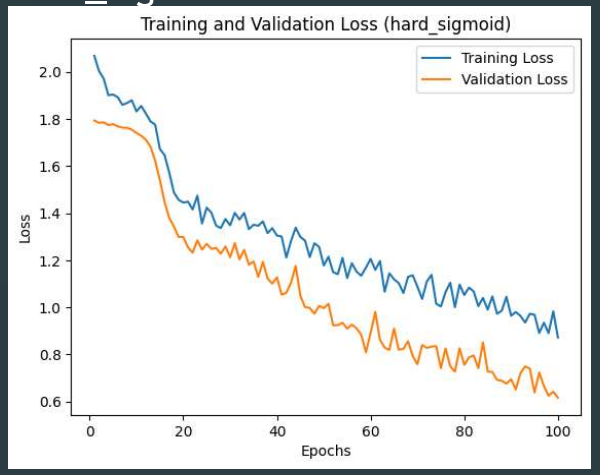
silu



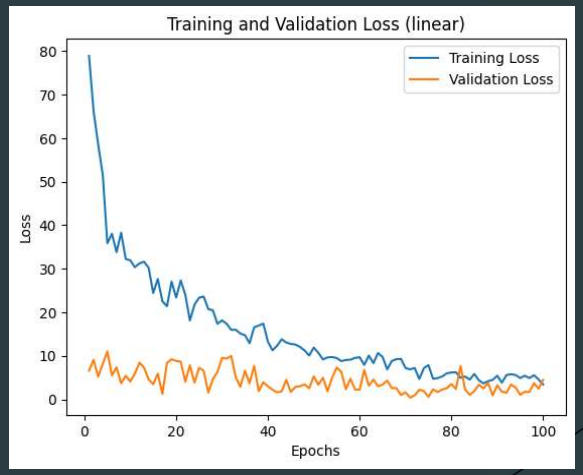
gelu



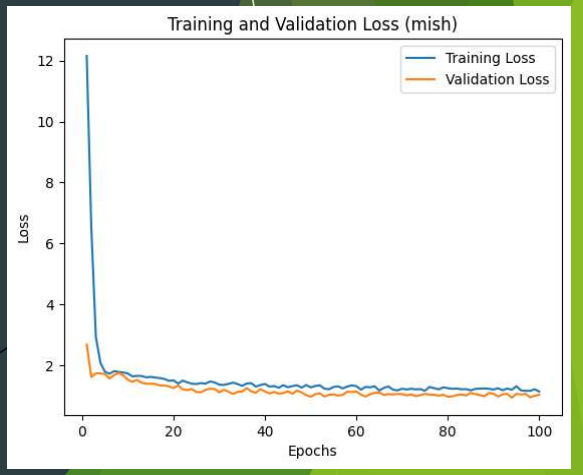
hard_sigmoid



linear



mish



Conclusio on Training and Validation Loss

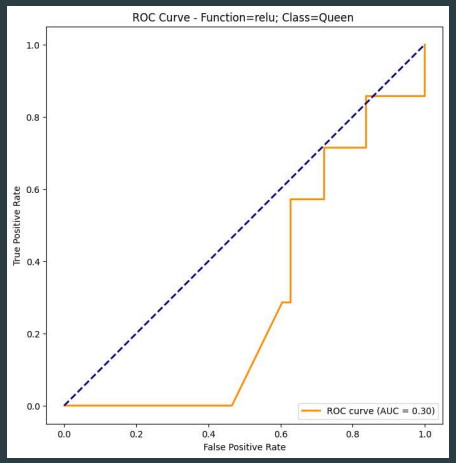
Sigmoid, tanh and hard sigmoid start with quite good loss results already at the first epochs.

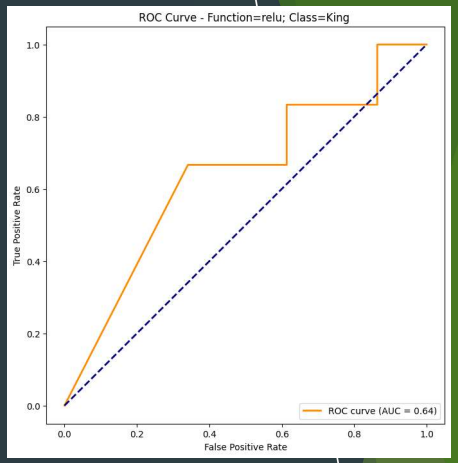
Nevertheless the other activation functions are catching up within the first 10 epochs and finetune at the rest of the epochs, so the results are comparable.

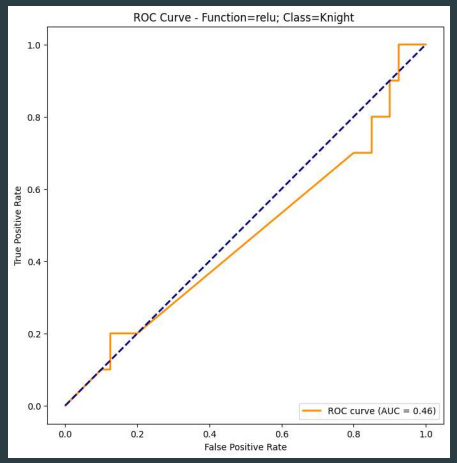
All functions seem to have even more potential when increasing the epochs a little bit more. Due to high computing time we had to reduce to 100 epochs for the run with all activation functions. When testing the epochs with relu we found that 200 epochs would fit best for our use case.

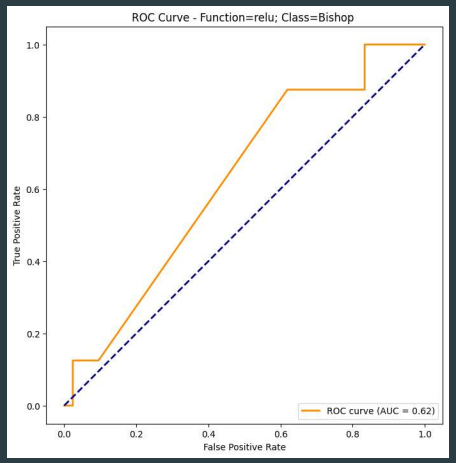
AUC (on validation set)

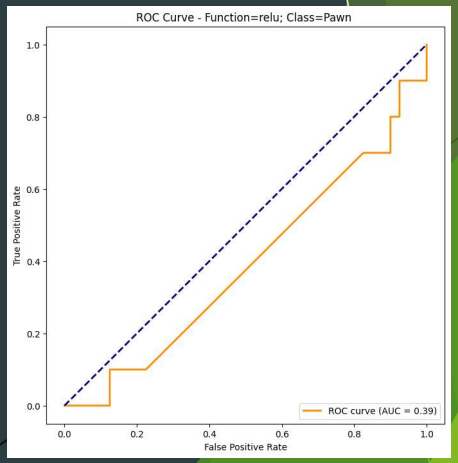
relu





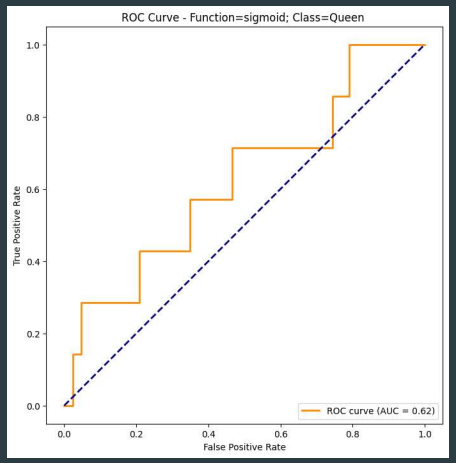


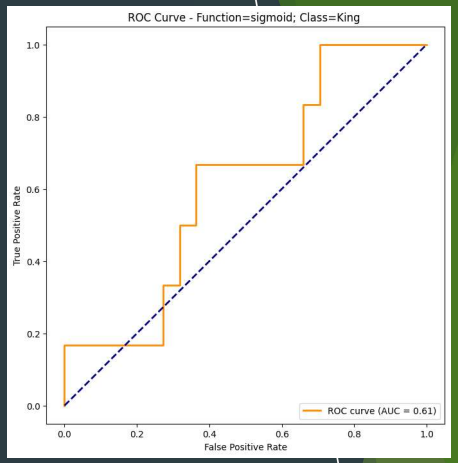


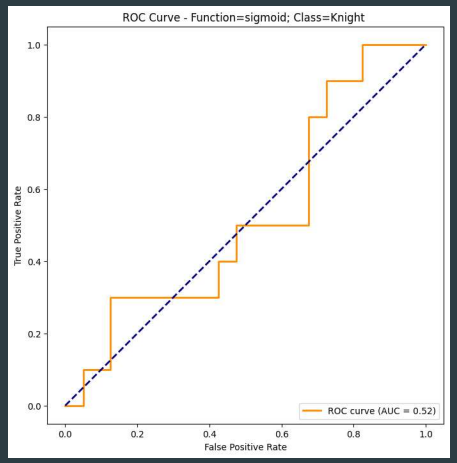


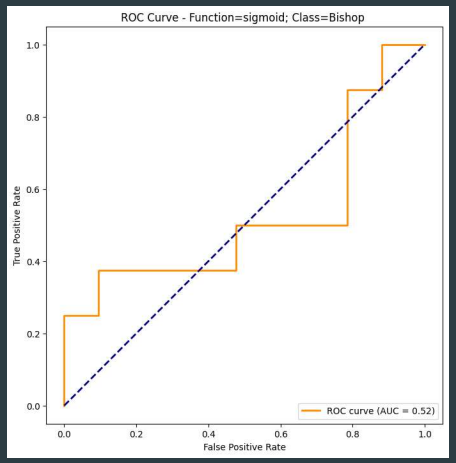
AUC (on validation set)

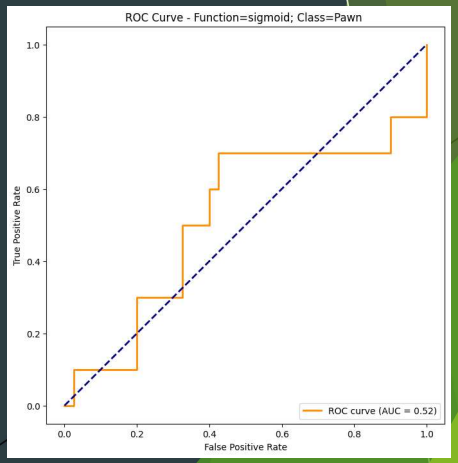
sigmoid





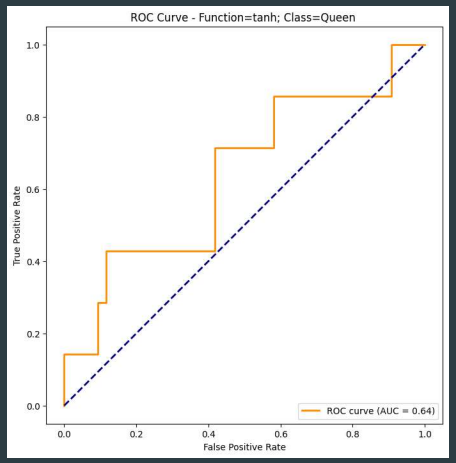


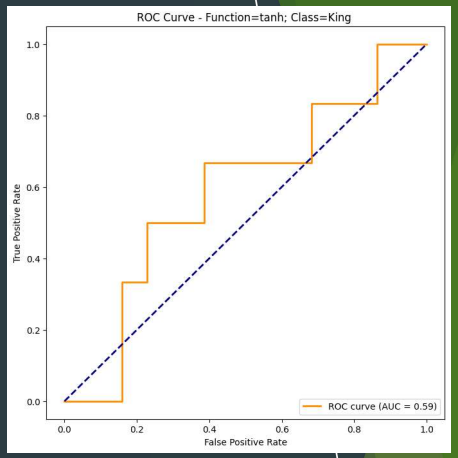


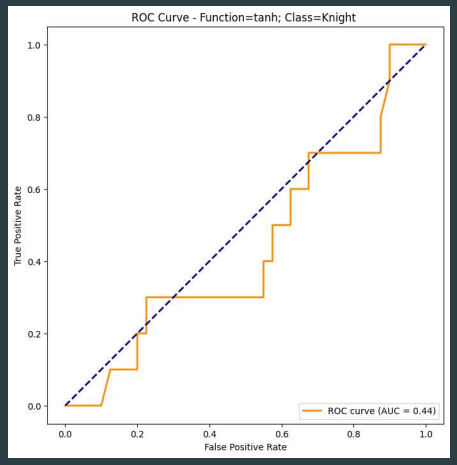


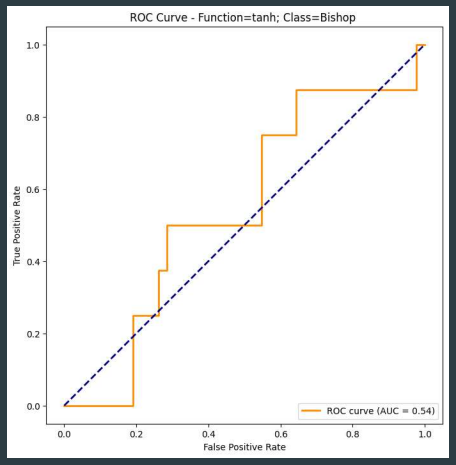
AUC (on validation set)

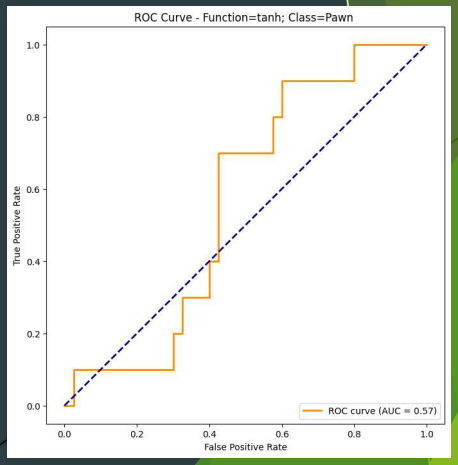
tanh





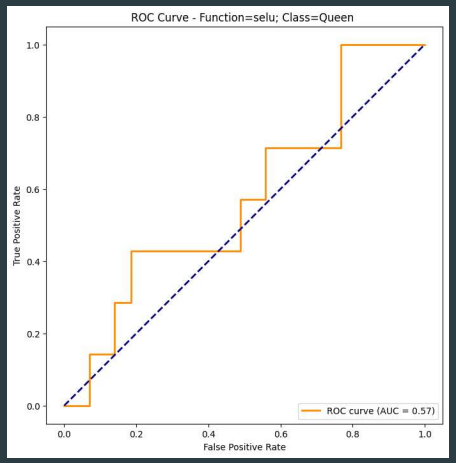


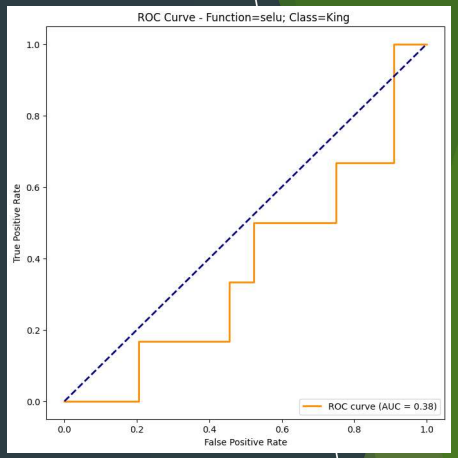


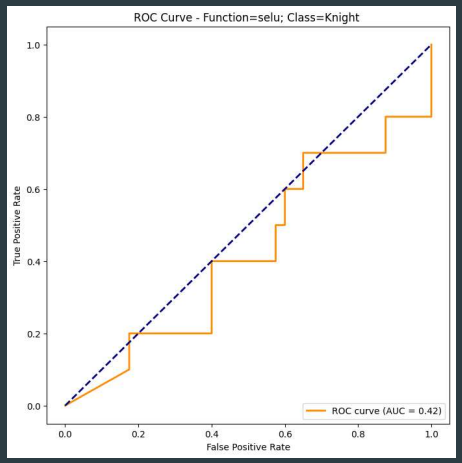


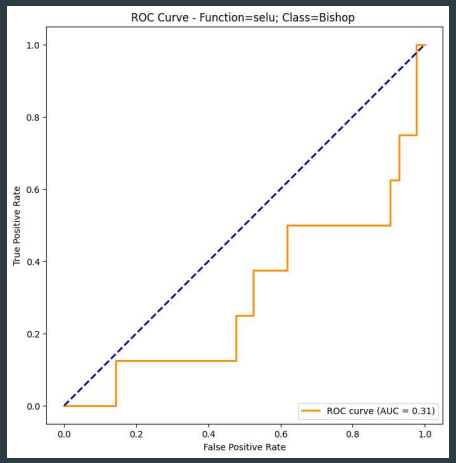
AUC (on validation set)

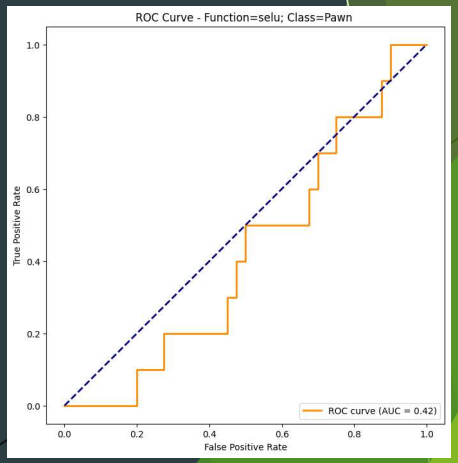
selu





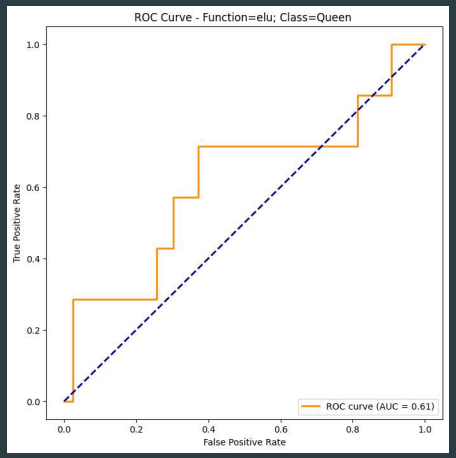


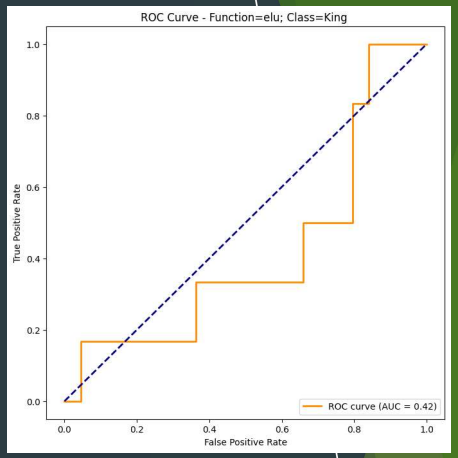


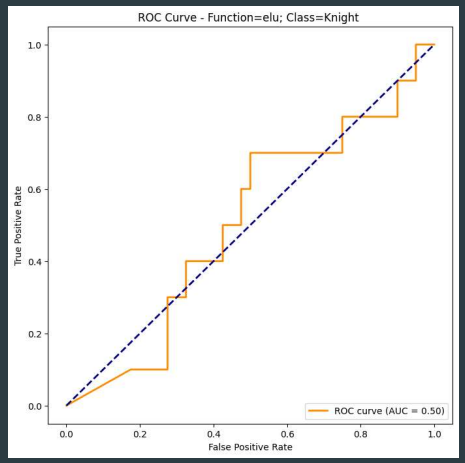


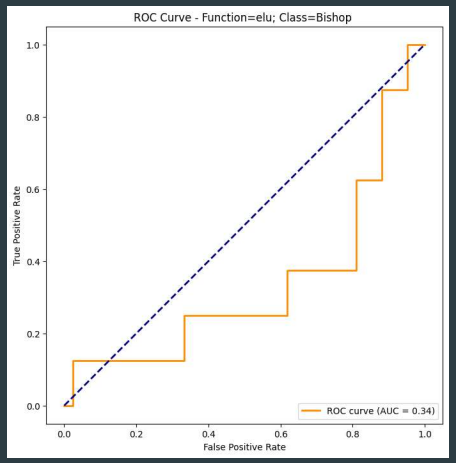
AUC (on validation set)

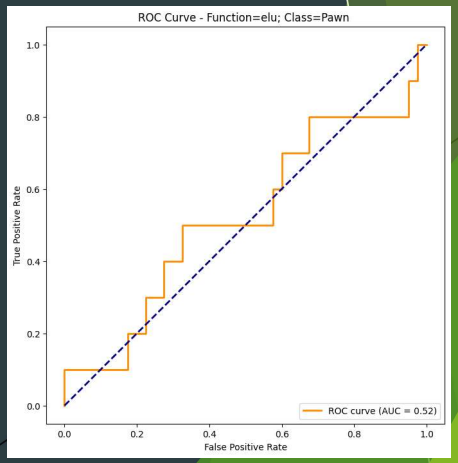
elu





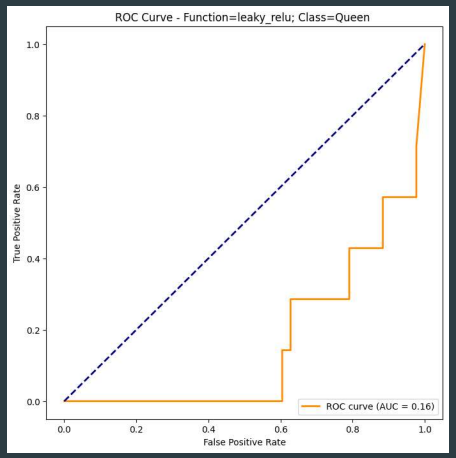


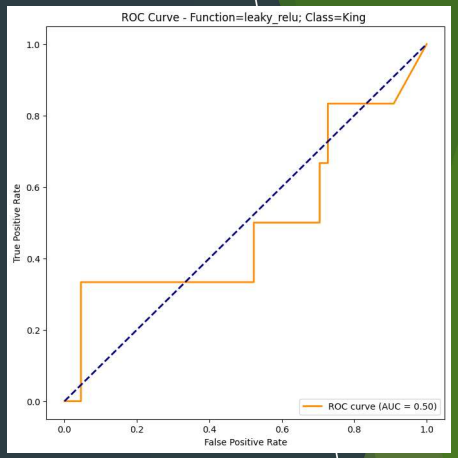


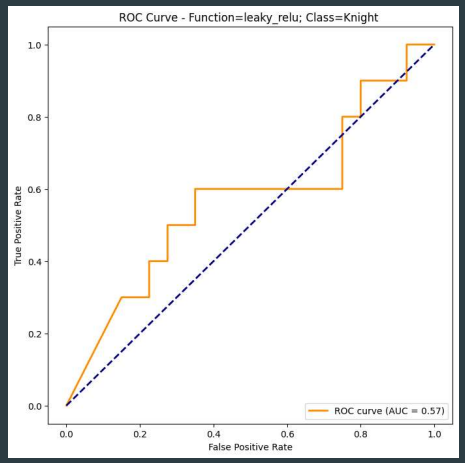


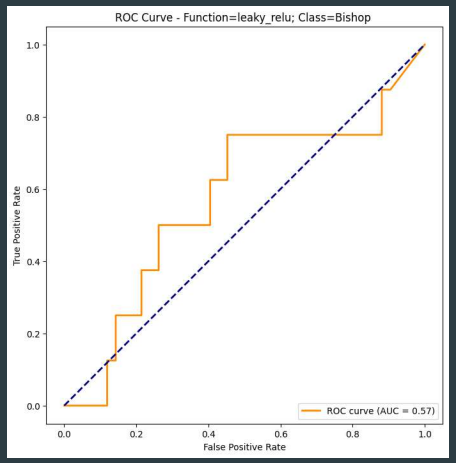
AUC (on validation set)

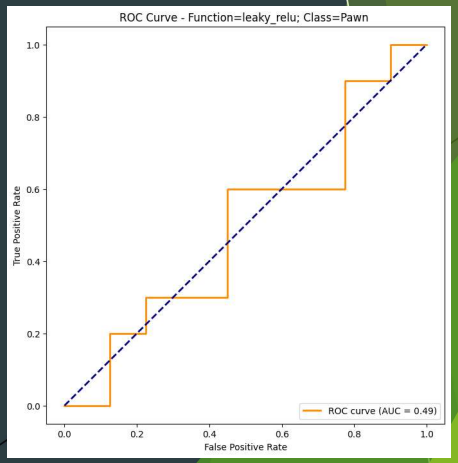
leaky_relu





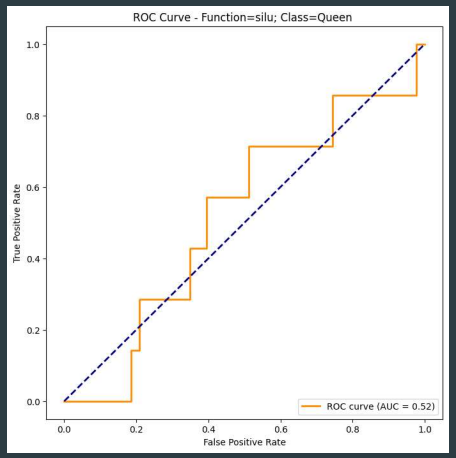


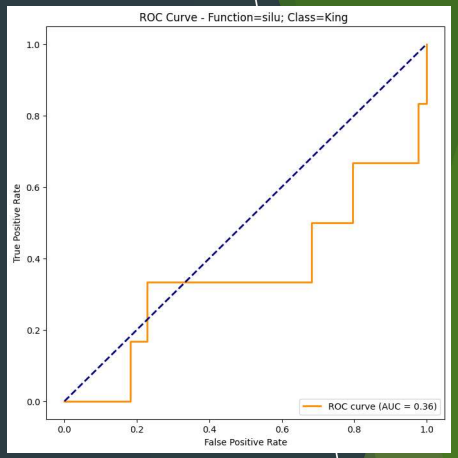


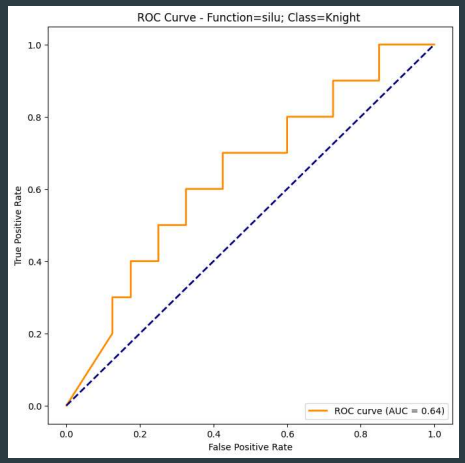


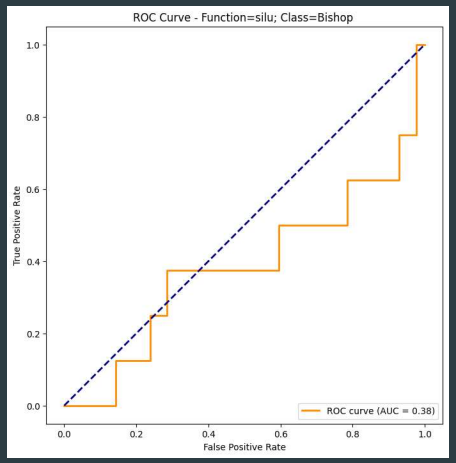
AUC (on validation set)

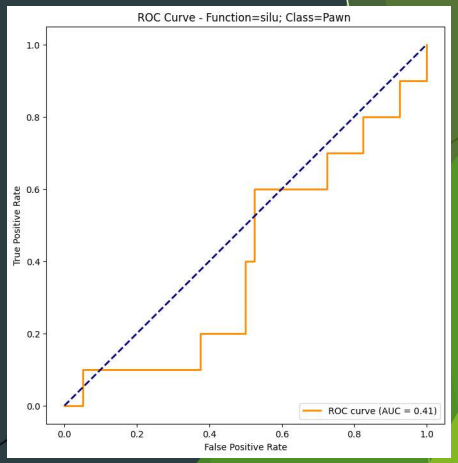
silu





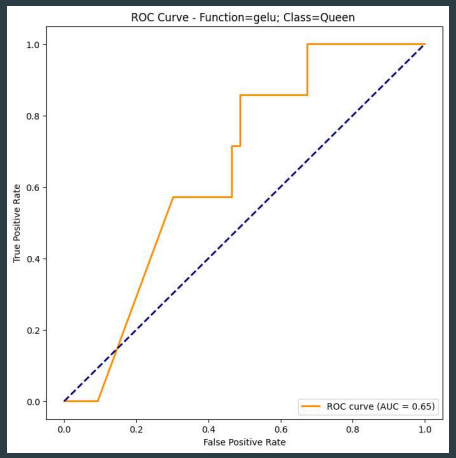


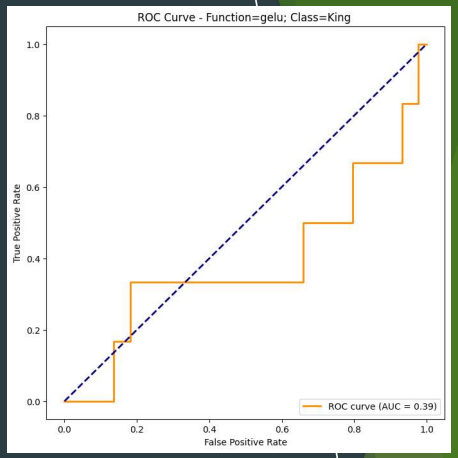


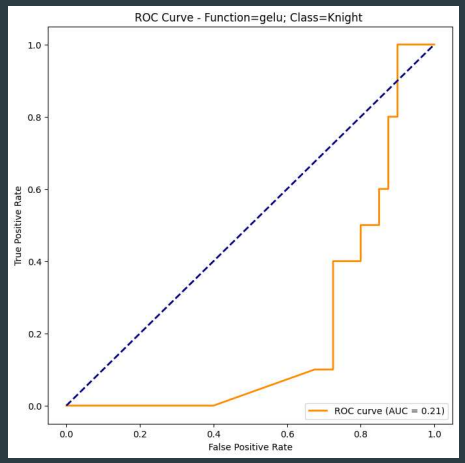


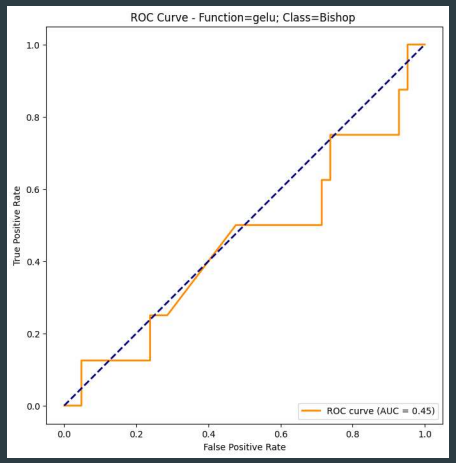
AUC (on validation set)

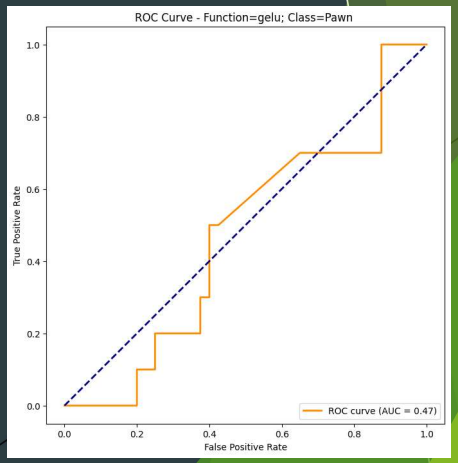
gelu





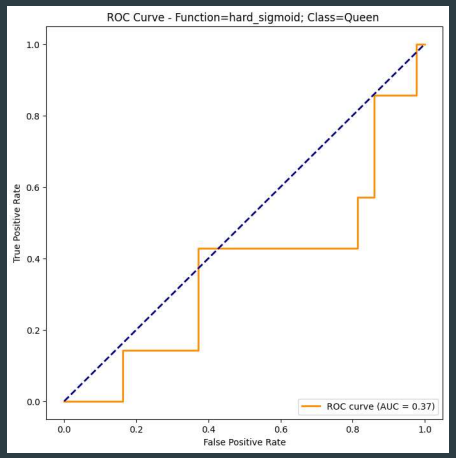


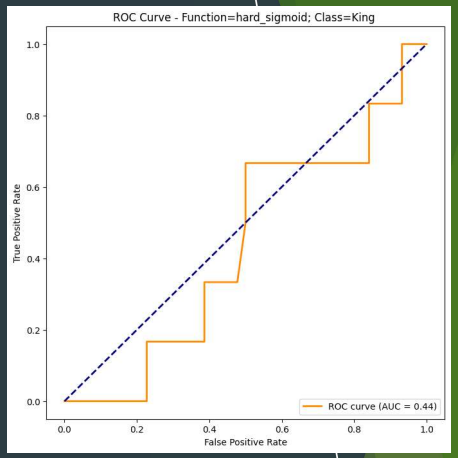


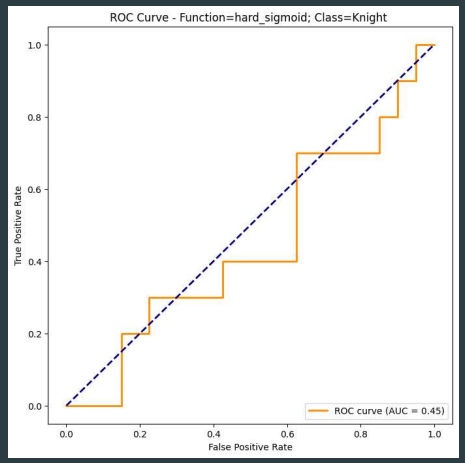


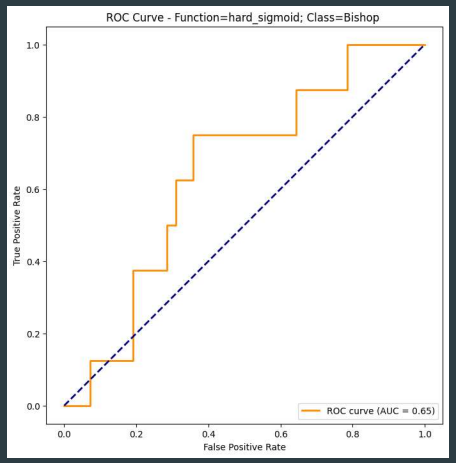
AUC (on validation set)

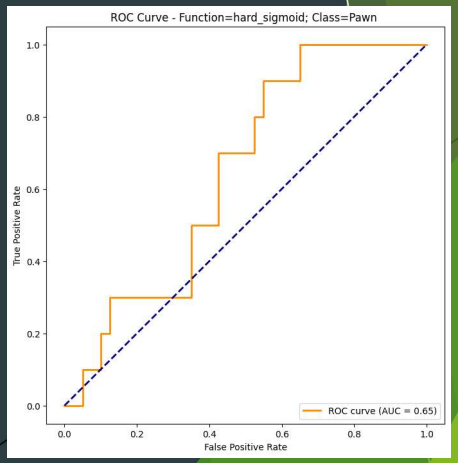
hard_sigmoid





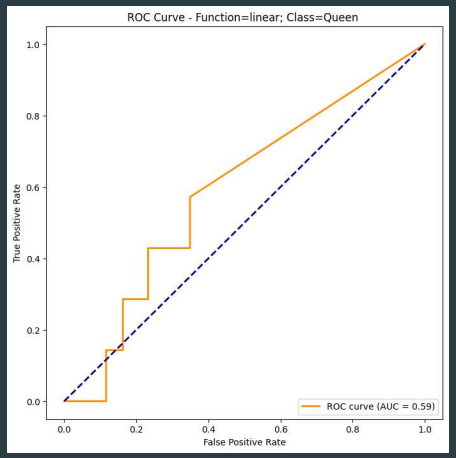


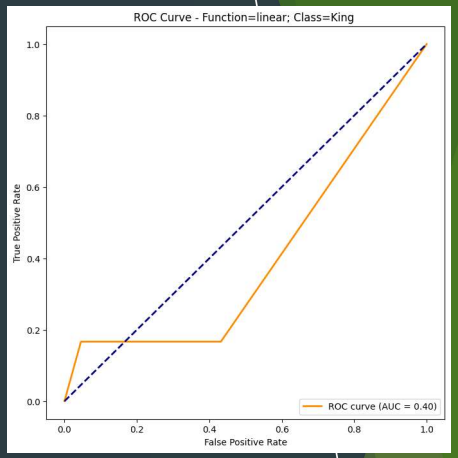


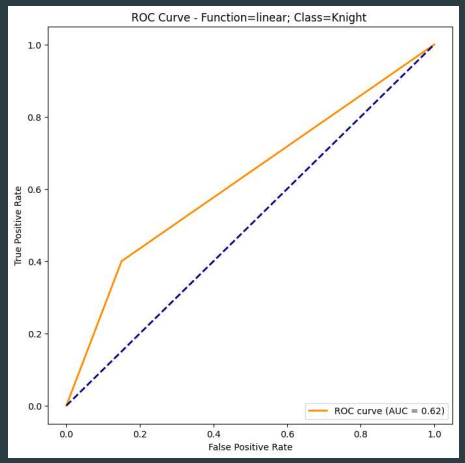


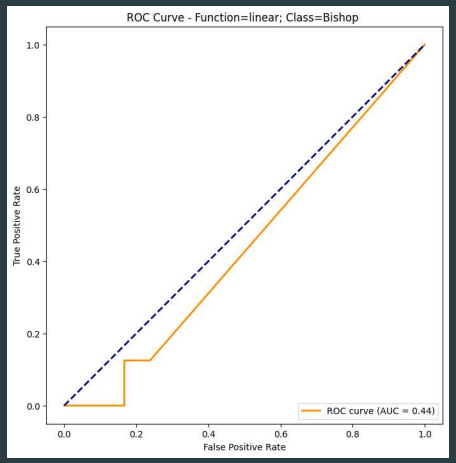
AUC (on validation set)

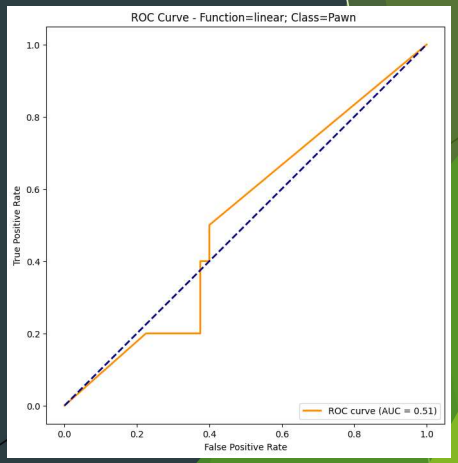
linear





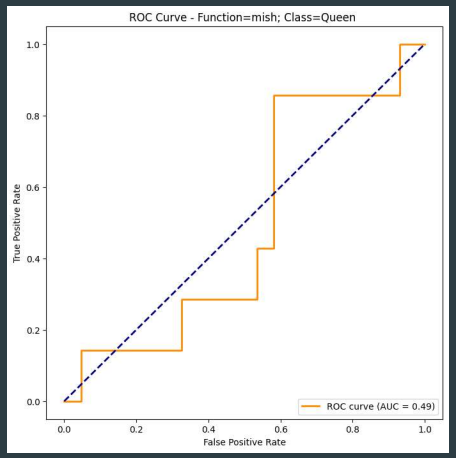


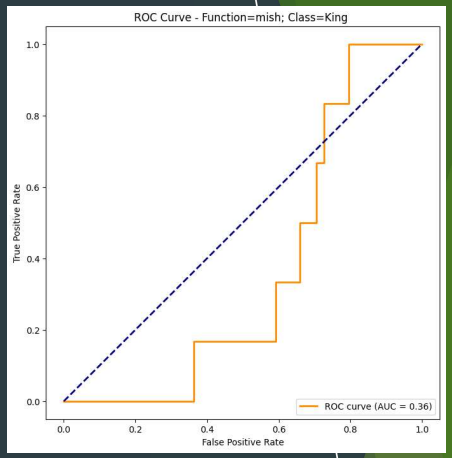


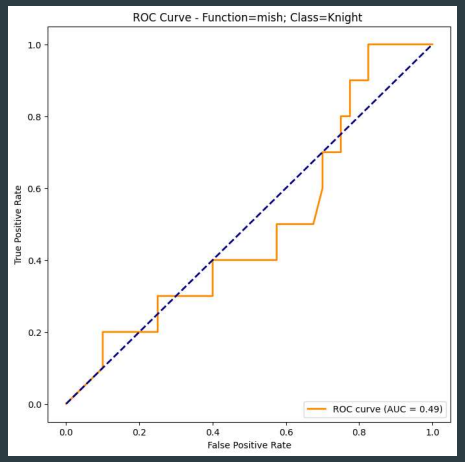


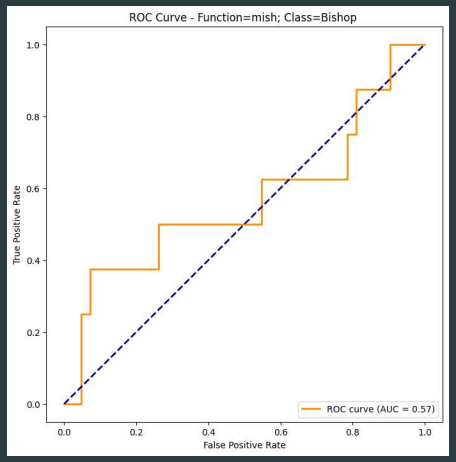
AUC (on validation set)

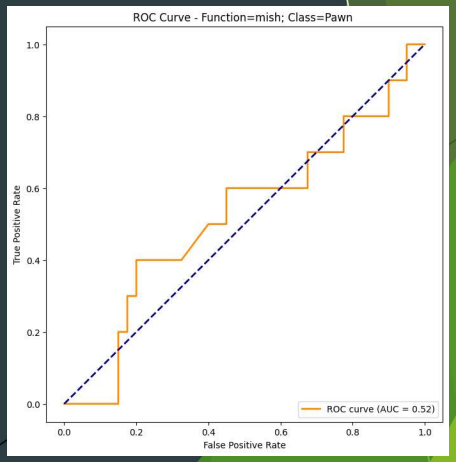
mish











Discussion on ROC Curves

- Overall bad results – will be discussed later
- Models were mostly around 0.5 for all classes
  - Probably random guessing the images
  - When one class was really good (>0.65), then another class really bad (<0.4)
- Sigmoid activation function was best
  - Average AUC over all classes 0.56
  - Above 0.5 in all classes

Running the best model (sigmoid) on test data

```
Actual class: Bishop, Predicted class: King
Actual class: Bishop, Predicted class: King
Actual class: Bishop, Predicted class: King
Actual class: Bishop, Predicted class: Rook
Actual class: Bishop, Predicted class: Rook
Actual class: King, Predicted class: Pawn
Actual class: King, Predicted class: Pawn
Actual class: King, Predicted class: Pawn
Actual class: King, Predicted class: Rook
Actual class: King, Predicted class: Rook
Actual class: King, Predicted class: Queen
Actual class: King, Predicted class: Queen
Actual class: Knight, Predicted class: King
Actual class: Knight, Predicted class: King
Actual class: Knight, Predicted class: Knight
Actual class: Knight, Predicted class: King
Actual class: Knight, Predicted class: Knight
Actual class: Pawn, Predicted class: Pawn
Actual class: Pawn, Predicted class: Rook
Actual class: Pawn, Predicted class: Pawn
Actual class: Pawn, Predicted class: King
Actual class: Pawn, Predicted class: Knight
Actual class: Queen, Predicted class: King
Actual class: Queen, Predicted class: King
Actual class: Queen, Predicted class: Pawn
Actual class: Queen, Predicted class: Pawn
Actual class: Queen, Predicted class: Queen
Actual class: Rook, Predicted class: King
Actual class: Rook, Predicted class: Knight
Actual class: Rook, Predicted class: Pawn
Actual class: Rook, Predicted class: Pawn
Actual class: Rook, Predicted class: Bishop
```

Conclusio

- The run of the best model on test set performed really badly
- Tries with different network topologies did not bring better results
- Possible reasons:
  - Some pictues contain multiple chess figures (left picture)
  - Very different styles for figures (middle two pictues)
  - The chessboard with high contrast edges might confuse the model