Tic-Tac-Toe CNN Model

Supervised By Dr.Mohamed Yousef

Gehad Omar

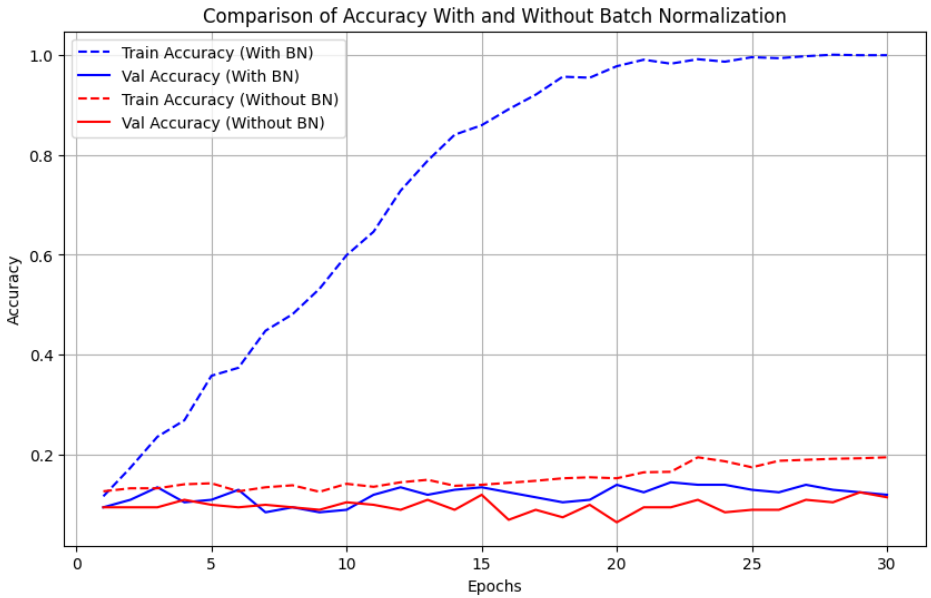
CS Department

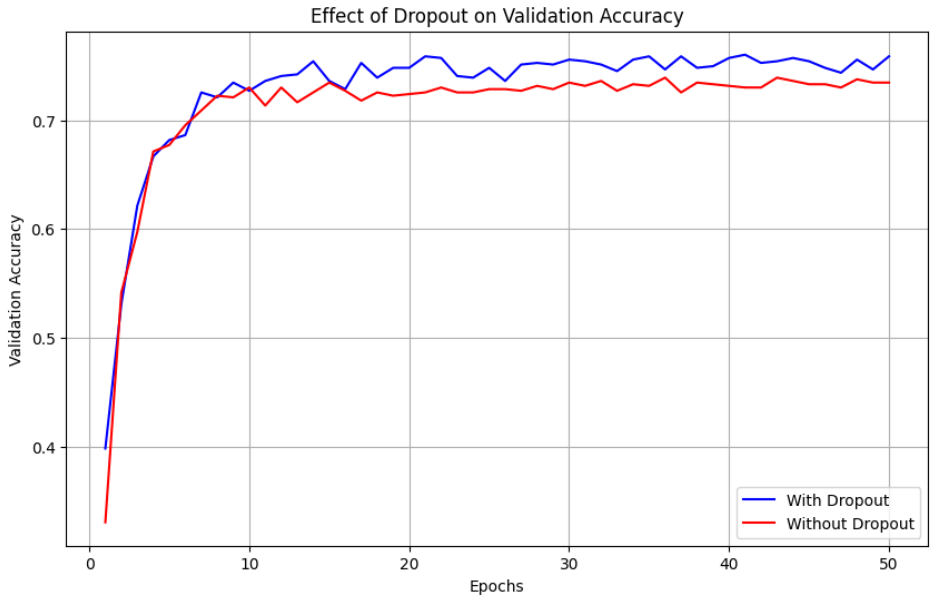
February 1, 2025

Model Architecture

- **Convolutional Neural Network (CNN)**:
  - 4 convolutional layers with increasing filter sizes (32, 64, 128, 256).
  - Batch Normalization and ReLU activation after each convolutional layer.
- **Dropout**:
  - Dropout layer with a rate of 0.5 to prevent overfitting.
- **Fully Connected Layers**:
  - Two fully connected layers to map features to 9 possible moves.

Regularization Techniques



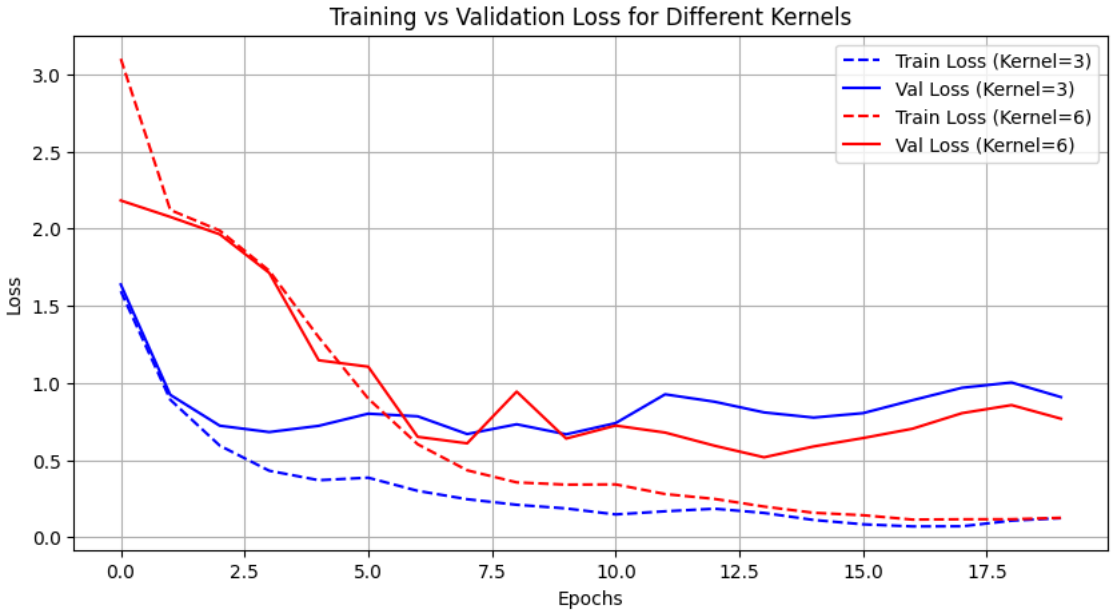


Model Architecture

- **Convolutional Neural Network (CNN)**:
  - 4 convolutional layers with increasing filter sizes (32, 64, 128, 256).
  - Batch Normalization and ReLU activation after each convolutional layer.
- **Dropout**:
  - Dropout layer with a rate of 0.5 to prevent overfitting.
- **Fully Connected Layers**:
  - Two fully connected layers to map features to 9 possible moves.

Kernel Size Comparison Study

- **Key Findings**:
  - Kernel=3 shows lower initial loss
  - Kernel=6 exhibits higher training volatility
  - Both converge but K=3 more stable
- **Implications**:
  - Smaller kernel better suited for Tic-Tac-Toe patterns
  - More efficient training process



Data Preprocessing

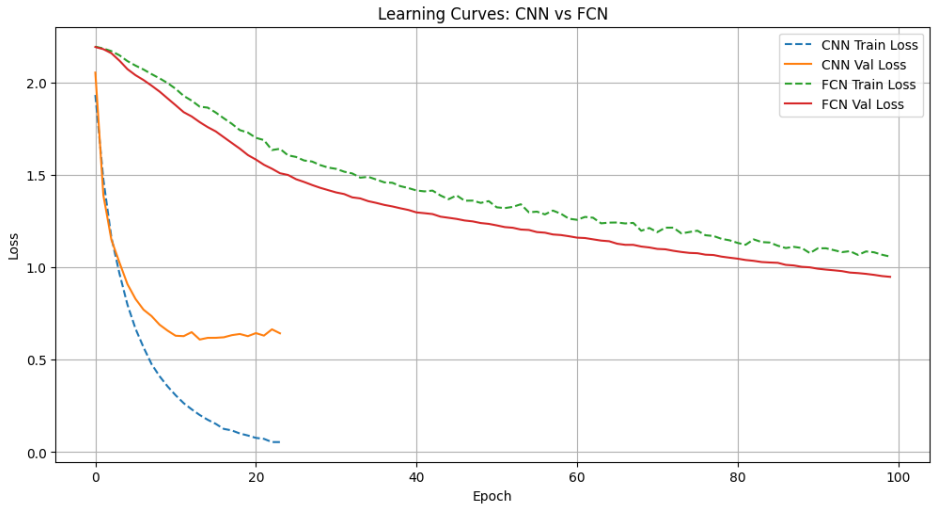
- **Board State Encoding**:
  - Convert board into 3 channels: current player, opponent, and empty spaces.
- **Data Cleaning**:
  - Remove duplicate board states to avoid redundancy.

Training Techniques

- **Minibatch Training**:
  - Dataset split into minibatches of size 128.
- **Loss Function**:
  - CrossEntropyLoss for multi-class classification.
- **Optimizer**:
  - AdamW optimizer with learning rate 0.0001 and weight decay 1e-4.
- **Learning Rate Scheduling**:
  - StepLR scheduler reduces learning rate by 0.1 every 10 epochs.
- **Early Stopping**:
  - Stops training if validation loss does not improve for 10 epochs.

Model Performance Comparison

- **CNN Model**:
  - Achieved accuracy: 81%
  - Superior spatial pattern recognition
- **FCN Model**:
  - Achieved accuracy: 69%
  - 12% lower performance



Evaluation Techniques

- **Validation Set**:
  - Dataset split into training (80%) and validation (20%).
- **Confusion Matrix**:
  - Evaluate model performance on validation set.
- **Classification Report**:
  - Provides precision, recall, and F1-score for each class.

Visualization Techniques

- **Learning Curves**:
  - Plot training and validation loss over epochs.
- **Bias-Variance Tradeoff**:
  - Plot bias, variance, and total error.

Interactive Gameplay

- **Human vs. Model**:
  - Allows human player to play against the trained model.
- **Win/Draw Detection**:
  - Functions to check game state for win or draw.

Regularization Techniques

- **Dropout**:
  - Used in fully connected layer to prevent overfitting.
- **Weight Decay (L2 Regularization)**:
  - Applied in AdamW optimizer to penalize large weights.

Model Saving and Loading

- **Model Checkpointing**:
  - Save the best model based on validation loss.
- **Loading Pre-trained Models**:
  - Load saved model for evaluation or gameplay.

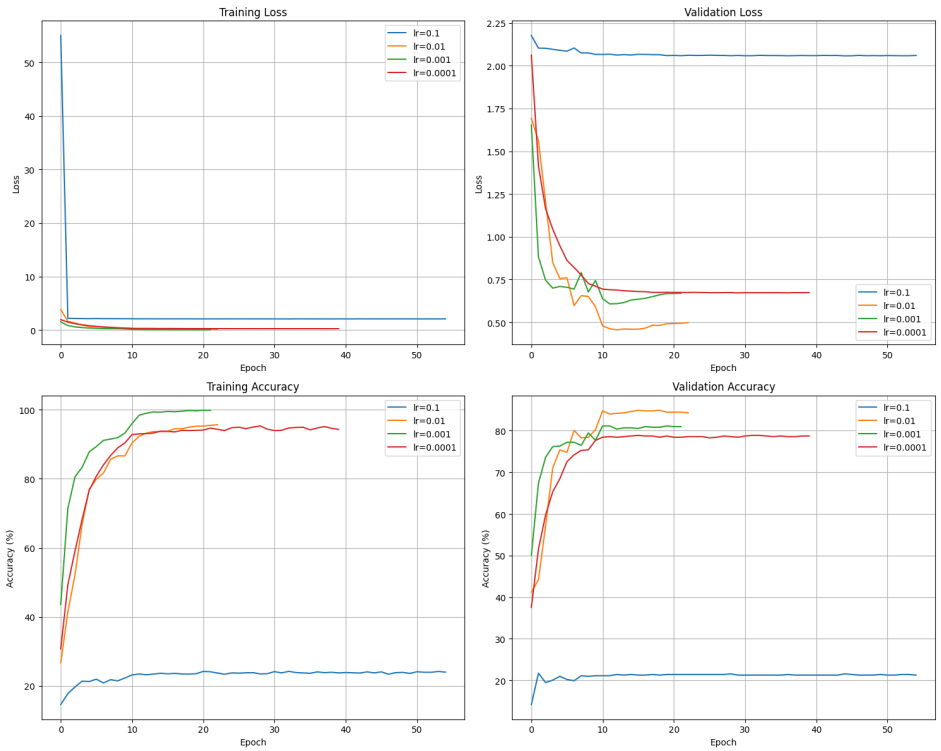
Training Techniques

- **Minibatch Training**:
  - Dataset split into minibatches of size 128.
- **Loss Function**:
  - CrossEntropyLoss for multi-class classification.
- **Optimizer**:
  - AdamW optimizer with learning rate 0.0001 and weight decay 1e-4.
- **Learning Rate Scheduling**:
  - StepLR scheduler reduces learning rate by 0.1 every 10 epochs.
- **Early Stopping**:
  - Stops training if validation loss does not improve for 10 epochs.

Learning Rate Analysis

Learning Rate	Train Acc (%)	Val Acc (%)
0.1	23.98	21.27
0.01	**95.66**	**84.31**
0.001	99.81	81.00
0.0001	94.30	78.73

- **Optimal Rate**: 0.01
- Best balance of training and validation accuracy
- Higher rates lead to instability
- Lower rates show overfitting signs



Dataset Handling

- **Combining Datasets**:
  - Concatenate multiple datasets (e.g., minimax, MCTS).
- **Dynamic Player Encoding**:
  - Preprocessing function encodes board based on current player.

Miscellaneous Techniques

- **Move Validation**:
  - Ensure only valid moves are predicted.
- **Early Stopping**:
  - Stop training if validation loss does not improve.

Model Comparison

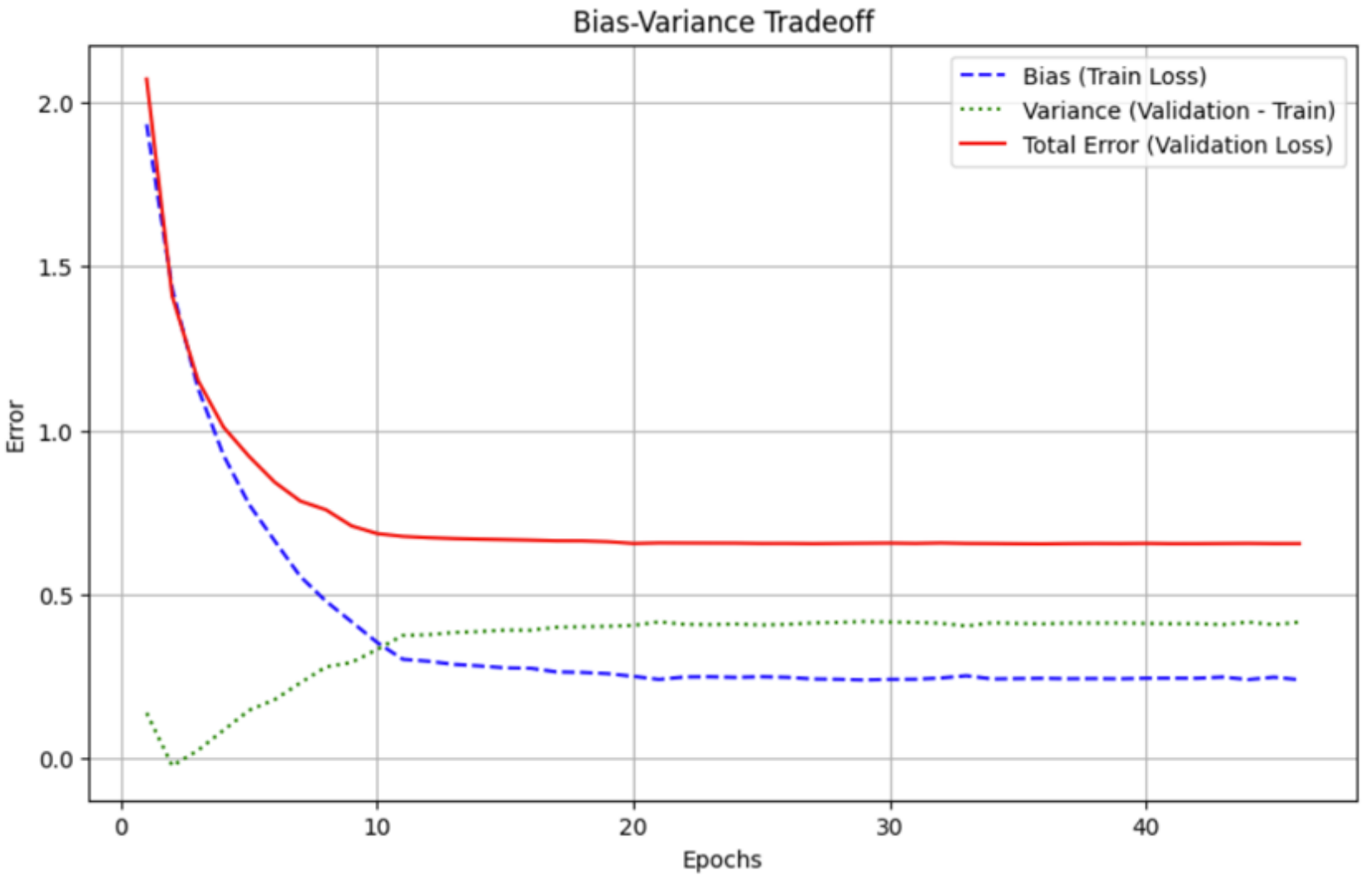
Model	Dataset	Accuracy
Model 1	Minimax	66%
Model 2	MCTS	67%
Model 3	Combined Rules	80%

Table: Comparison of Models with Different Datasets

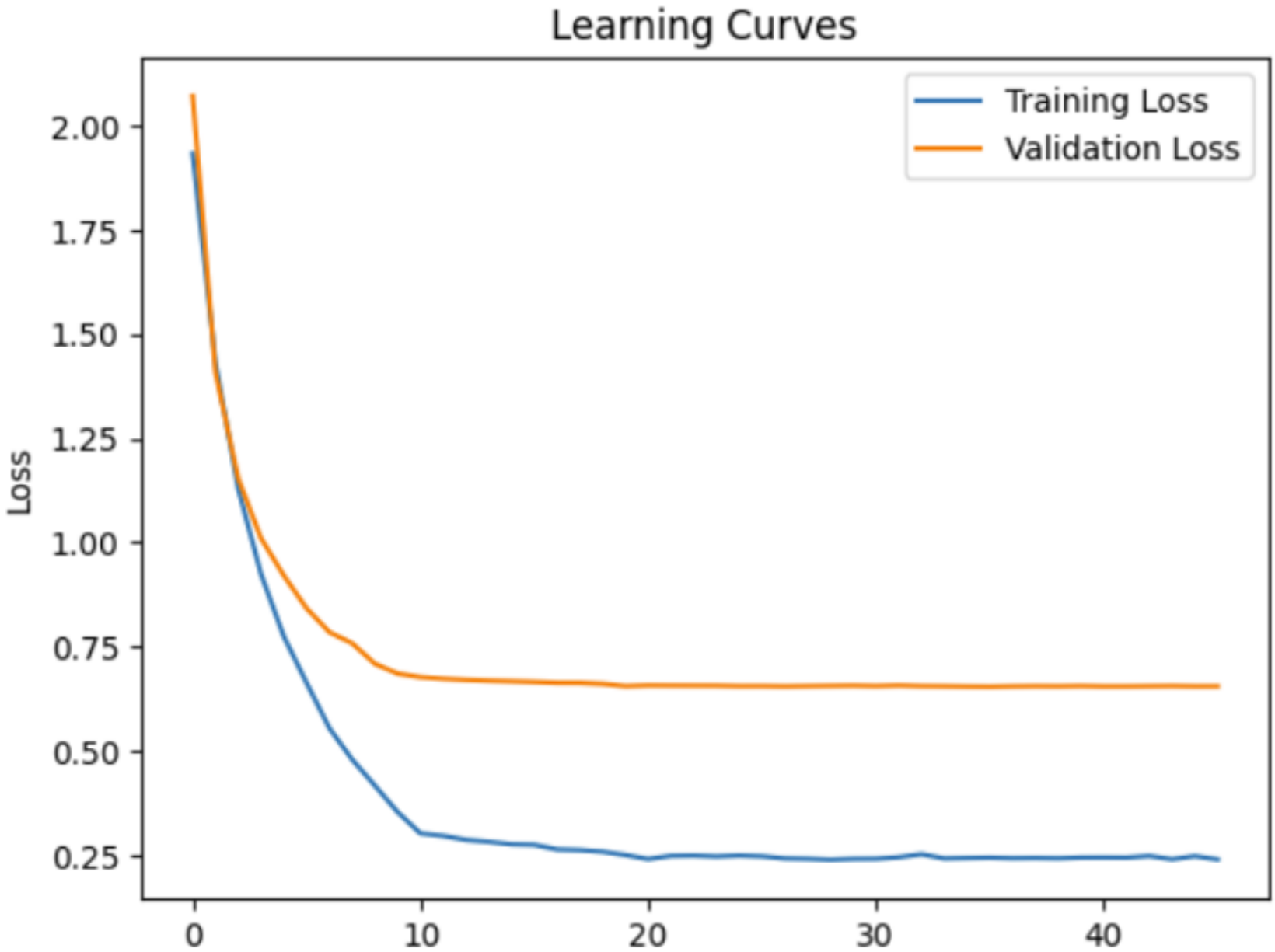
Confusion Matrix

```
Confusion Matrix:
[[ 83   1   5   0  11   1   0   1   2]
 [  1  50   3   0   0   0   1   0   2]
 [  3   0  83   0   9   1   3   0   1]
 [  0   1   1  42   0   0   2   0   0]
 [  6   0   5   0 104   1  10   0  11]
 [  1   0   1   1   2  30   3   0   0]
 [  8   1   5   0   6   0  55   0   6]
 [  0   2   0   0   0   0   0  22   0]
 [  2   0   2   4   6   0   3   0  59]]
```

Bias-Variance Tradeoff



Learning Curves



Conclusion

- The CNN model achieved 81% accuracy in predicting Tic-Tac-Toe moves.
- Smaller kernel sizes (e.g., 3x3) were more effective for this task.
- Explore the full project on Google Colab: `https://colab.research.google.com/drive/1OsUM5DhCCEcu_utApdhz_bIMLE6_Yr8S#scrollTo=FRYzH1tn3jyr`