Predicting the Outcome of a Pass Play Using Convolutional Neural Network

Elizabeth Howell | Utah State University

Introduction

"When a quarterback takes a snap and drops back to pass, what happens next may seem like chaos. As offensive players move in various patterns, the defense works together to prevent successful pass completions and then to quickly tackle receivers that do catch the ball."

- *Excerpt from NFL Big Data Bowl Kaggle Competition*

The NFL challenged participants to "use data science to better understand the schemes and players that make for a successful defense against passing plays". Because there is not objective goal, I am able to create my own problem. Creating an accurate model that can predict pass results based on player position would benefit both offence and defence by allowing coaches adjust player placement that provide the best possibility of the desired outcome. An example for offence it would be a pass completion, or for defence it would be a QB sack, an interception, or a blocked pass.

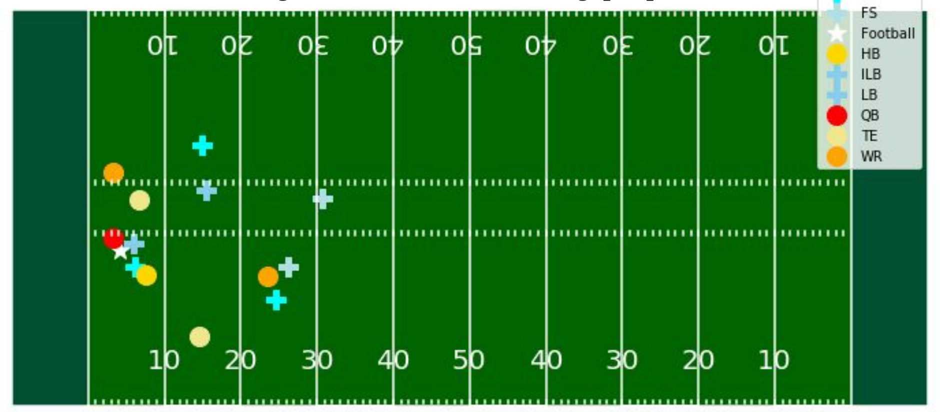
Data

Figure 1: Columns Utilized from Provided Data Sets

Data Set	Data Description	Columns	
Play Data	File plays.csv contains play-level information from each game.	PlayID	Play Identifier
		GameID	Game Unique Identifier
		PassResult	Outcome of the passing play (text) - C: Complete pass - I: Incomplete pass - S: Quarterback sack - IN: Intercepted pass
Tracking Data	Files week[week].csv contain player tracking data from all games in week [week]. The tracking data provided the each player position for every player, besides the offensive line, and the football of the field for each second of a play.	PlayID	Play Identifier
		GameID	Game Identifier
		FrameID	Frame Identifier for each play starting at 1
		X	Player position along the long axis of the field, 0 - 120 yards (numeric)
		Y	Player position along the short axis of the field, 0 - 53.3 yards. (numeric)
		Position	Player position group (text)
		Event	Tagged play details, including moment of ball snap, pass release, pass catch, tackle, etc (text)

Using the provided information, I created an algorithm that creates images that represent player positions on a football field right before quarterback passes the ball. If the quarterback fails to pass the ball, the image represents the frame after the ball has been snapped. Plays where the play result is not available are omitted. The image name is a combination of the playID and gameID. A CSV file holds the image name and the play result.

Figure 2: Example: PlayID: 210 GameID: 2018090902 - Play Result: C (Complete)
Legend included for viewing purposes



The training data contains approximately 17,000 images and the test data contains 2,500 images. The algorithm, and the link to the images, and csvs are included repository.

Goal:

Predict the outcome (Completed, Incomplete, Sack, or Interception) of a pass play based on player positions using a Convolutional Neural Network.

Approach

Pre-Processing:

A custom data set was created to handle the images and labels. Each data item returns the image, the play result, and item id (the playid and game id). Before item is returned, the image is resized to 56x56 and normalized, and the the play result is mapped to an integer. (C:1, I:2,S:3, IN:4).

Neural Network

The network is comprised of a single pretrained ResNeXt-50(32x4d) CNN with the following modifications:

- The first convolutional layer is initialized using a 3x3 kernel size, 2x2 stride, and 1x1 padding
- The max pool is changed to an Identity layer
- The fully connected layer (fc) is a sequential layer with the following layers:
  - A Dropout layer
  - A Linear Layer that outputs 10.

Loss Function: Cross Entropy
Optimizer: Adam Optimizer with a weight decay and learning rate of 1e-5
Learning Rate Scheduler: StepLr with step size = 100 and gamma = .01

Experiments

1. Prediction Accuracy:
The prediction accuracy was calculated by the number of correct predictions divided by total number of predictions. In addition, prediction accuracy is calculated for each pass result.

2. Compare Models:
An additional ResNeXt-50(32x4d) model with no additional modifications was also trained and tested with the same data. The training and test results were then compared to the results modified ResNeXt-50(32x4d) model. The structure of the unmodified ResNeXt-50(32x4d) model can be viewed in Figure 3.

stage	output	ResNeXt-50 (32×4d)
conv1	112×112	7×7, 64, stride 2
conv2	56×56	3×3 max pool, stride 2 [1×1, 128; 3×3, 128, C=32; 1×1, 256] ×3
conv3	28×28	[1×1, 256; 3×3, 256, C=32; 1×1, 512] ×4
conv4	14×14	[1×1, 512; 3×3, 512, C=32; 1×1, 1024] ×6
conv5	7×7	[1×1, 1024; 3×3, 1024, C=32; 1×1, 2048] ×3
	1×1	global average pool 1000-d fc, softmax
# params.		**25.0**$\times 10^6$
FLOPs		**4.2**$\times 10^9$

Figure 3

Links & References:

Kaggle Competition: https://www.kaggle.com/c/nfl-big-data-bowl-2021/overview
Source Code: https://github.com/erhowell/CS5665Project

References:
Resnet 34 Image: https://pytorch.org/hub/pytorch_vision_resnext/
Football Field Generation Code: https://www.kaggle.com/robikscube/nfl-big-data-bowl-plotting-player-position

Results & Analysis

Figure 4

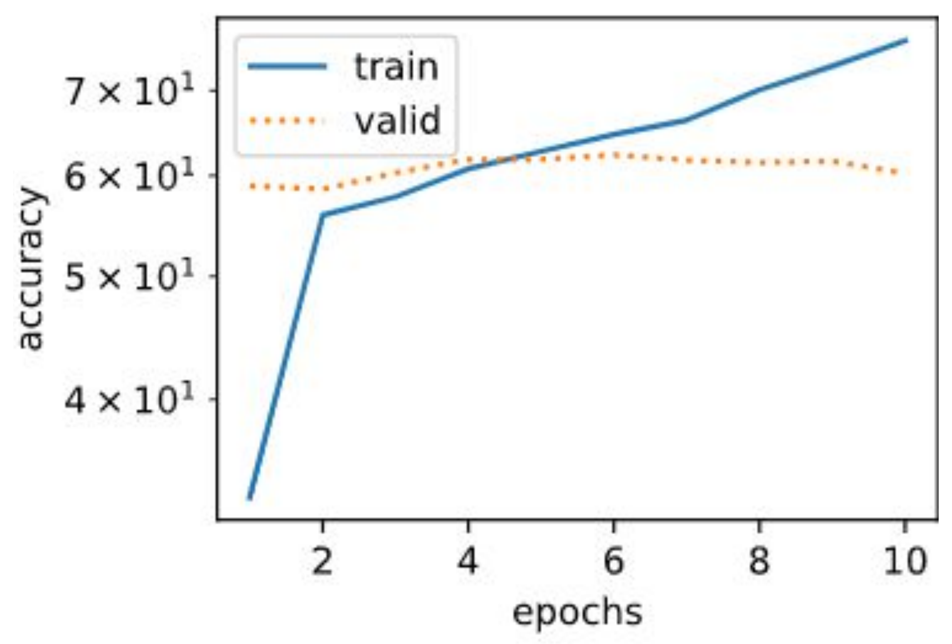


Figure 4 is a plot of the training and validation accuracy for each epoch. The little amount of change in validation accuracy suggests some over fitting.

Figure 5 represents the comparison between the distribution of each pass result class in the test dataset and the correct output predictions made by the ResNetxt model. Using the information from this model, I conclude I need to train the model for interceptions and incomplete passes. An explanation of the results is the small and unbalanced dataset. As you can tell from the graph , completed passes makeup almost 60% of the outputs, whereas interceptions make up about 2% of the data. In addition, my dataset was made up of less than 20,000 images, so the network didn't have enough images of interceptions to use for training. This results the neural network to be over trained, leading to inaccurate predictions. Some suggested fixes could be:

- Add more data that represents incomplete and intercepted passes
- Reduce the number data that represents a completed pass.

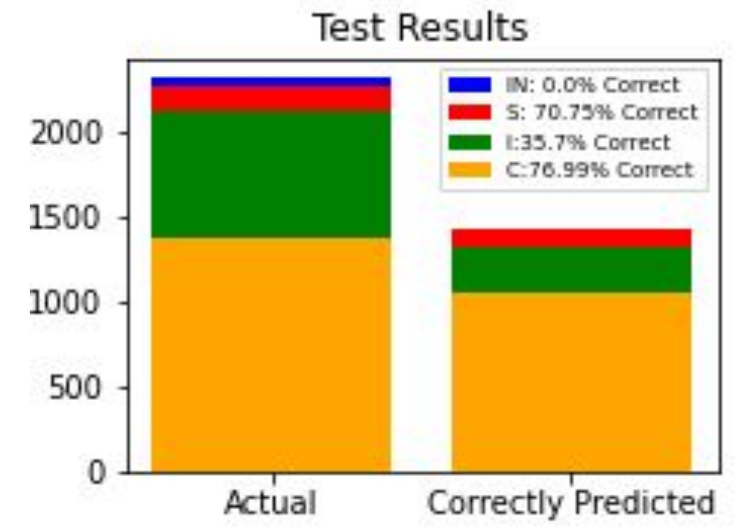


Figure 5

Model Comparison Results (Figure 5)

		ResNeXt-50 (32x4d)	**Modified** ResNeXt-50 (32x4d)
Train Results	Highest Training Accuracy	59.23%	76.41%
	Avg Training Accuracy	56.63%	62.14
	Highest Validation Accuracy	58.05%	63.24%
	Avg Validation Accuracy	57.37%	60.86
Test Results	Accuracy	57.56%	62.7%

Looking at the results in figure 5, the modified ResNeXt model outperforms the original model every time and has a higher test; however, the Modified Resnet model accuracy could be improved significantly. The improvements to the dataset stated previously would help the accuracy.

Conclusion

For this project I was able to create an algorithm that can create images that represent player positions on a football field, and I was able to create a CNN that can accurately predict Sacks and Pass Completions. This Kaggle competition does not score or rank submissions.
My next two goals to improve the model are:

- Create a larger and more balanced dataset.
- Analyze the distribution of incorrectly predicted output data for each of the pass result classes to determine how the network is getting the wrong output