# RouteNet: a convolutional neural network for classifying routes

Nathan Sterken
nathan.sterken@gmail.com
Seattle, WA

## Abstract

I assigned route labels to ~1,000 location traces of eligible receivers and then used that data to train a convolutional neural network (‘RouteNet’) to classify routes. I classified all ~35,000 receiver location traces in the Big Data Bowl dataset using RouteNet. Based on these labels, the game situation, and the positioning of the eligible receivers (pre-snap and at the time of ball arrival) this paper analyzes the effectiveness of various receiver-route combinations.

## Introduction

My goal with this paper is to respond to the third theme of the 2019 Big Data Bowl -- identifying the best receiver-route combinations. As an All-Pro armchair quarterback I knew what the answer should be before starting -- throw it deep, of course -- but I wanted to see if the tracking data backed up this belief. Specifically, to identify the best receiver-route combinations I built a system for classifying all routes, appended related datasets, computed additional variables, and then generated plots and regressions of my target variable. The rest of this paper explains those steps in more detail and presents the optimal receiver-route combinations by game situation.

## Defining successful outcomes for receivers

I use win probability added (WPA) as defined in the nflWAR paper [5] as my measure of success. WPA is great because (1) it distills each team’s goal into a single metric; (2) it accounts for game situation (for example, by recognizing that a 5-yd completion on 4th and 2 is great while a 5-yd completion on 4th and 6 is terrible, and that completions in garbage time actually aren’t worth much); and (3) it is easily available thanks to the great nflscrapR-data github repo [4]. The optimal receiver-route combinations are those that help the team using them win the game; as such, WPA is the best metric to evaluate receiver-route combinations.

That said, WPA does have a significant limitation: as a results-based metric that combines every player’s contributions into a single number it can ignore great individual performances that didn’t impact the play -- if a receiver gets wide open but the quarterback doesn’t see him, for example, WPA won’t give that receiver any credit. I considered building an alternate metric like ‘yards to nearest defender at the time of the pass’, as that might help isolate receiver-specific positive contributions, but ultimately decided against it; if a receiver is wide open but doesn't get the ball that would be difficult to interpret based on the tracking data alone -- he might be open because he is away from the play or is being left open intentionally (like when defenses try to bait an underneath pass in the two-minute drill). Still, this is a ripe area for future work (read: I look forward to reading the other entrants in this challenge).

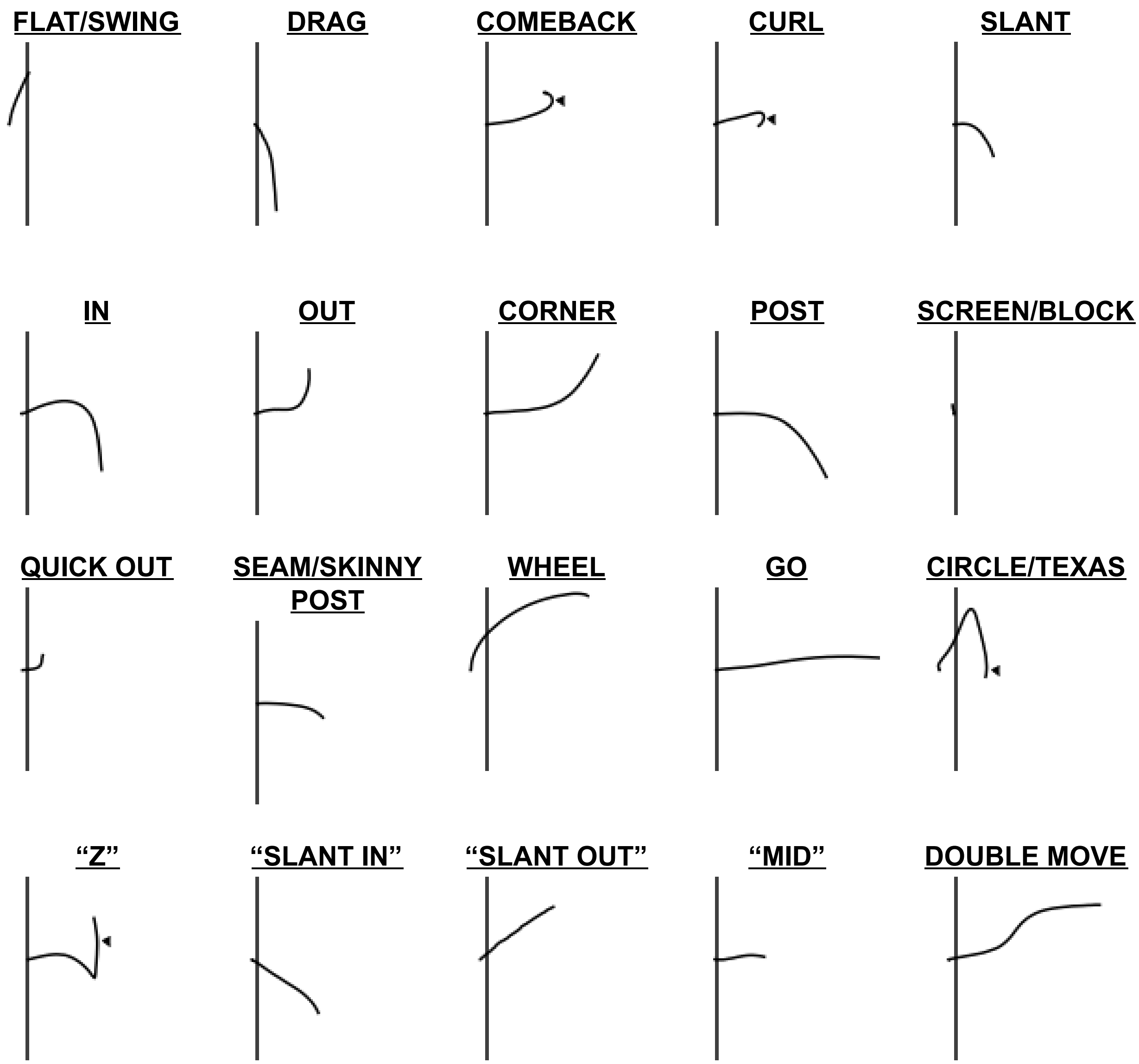
## Classifying routes

Inspired by the work of Hochstedler and Gagnon [3], I decided to build a supervised model for identifying routes. Rather than manually curate features, however, I decided to frame the task as an image recognition problem and generate features automatically with a convolutional neural network. Specifically, I assigned labels to each of the ~35,000 eligible receiver location traces in the data using the following process:

1. I plotted each passing play for three week one games (see the 'Full Play' plot in the Appendix for an example) and then assigned one of twenty labels to each route. This gave me ~1,000 labeled routes. When plotting the plays I stopped each play when the ball arrived (or a similar route-ending outcome happened, such as a sack) to avoid the noise introduced by movement after the catch. The full list of routes is explained in more detail below and the 'RouteNet Confusion Matrix' and 'RouteNet class validation plots' in the appendix shows how the network performed at classifying each type of route.
2. I generated route-specific images that would be suitable for input into a convolutional neural network. Specifically, I flipped the X and Y data as appropriate so that all routes were from the perspective of the receiver being 'above' the QB and moving to the 'right' of the image, I centered the images, created them at a uniform size in grayscale, and added a marker to indicate the first change of direction in the 'x' dimension. Then I assigned each image the label generated in step 1.
3. I split the labeled images into 'test' and 'training' sets based on the last digit of their playId -- plays ending in 0 and 1 were used for testing; the rest were used for training.
4. I trained a convolutional neural network to identify the labels. I used an architecture that has been proven to be successful for recognizing handwritten digits. Specifically, RouteNet consists of two convolutional layers and two fully connected layers, along with two dropout layers and ultimately a softmax layer.
5. I used the newly trained network to obtain interim labels for all routes from Week 2.
6. I reviewed the Week 2 labels assigned by the model and adjusted the labels as necessary. This review was much quicker than the original play tagging, since the plays had already been somewhat grouped by the network, but because I was reviewing plots that had been designed for the network as opposed to the full-play plots I originally used for classification it was also more error prone -- it was difficult to properly identify all routes without full context, so there is a fair amount of noise in the labeled data. I hoped to go back and re-classify more of the routes within the full context of the play but ran out of time. Still, this provided ~6,000 additional labeled examples and the validation plots and confusion matrix in the appendix demonstrate the high accuracy of the model.

7. I re-trained the model with all ~7,000 examples and then applied the updated network on all ~35,000 routes in the Big Data Bowl data set to obtain a label for each route.

I tagged twenty types of routes during my manual review. Canonical examples of each type of route are displayed below (the ball is always below the receiver's starting point):




Most of these routes are pulled from a standard route tree but a few merit additional discussion:

- **"Mid"** - I created this classification as a kind of grab bag for routes that were ~10 yards straight down the field without any discernible break -- these are likely actually longer routes that didn't have time to fully develop.
- **"Slant In/Out"** - I created these classifications to cover routes that ran at 45 and 135 degrees from the line of scrimmage. These seemed to be not a slant, not yet a corner; as noted below, however, they were combined with similar routes during analysis.

- **“Seam/Skinny Post”** - I created this category to catch straight mid-length routes that didn’t quite seem to be of ‘Go’ length. During analysis I grouped these with ‘Go’ routes.

I welcome any comments on these labels and if other classifications would be more appropriate; with that guidance it would be straightforward to adjust and re-train the model.

Some of the routes were rare so for the purposes of the receiver-route combination analysis I combined them with their most similar route. With more data it would likely be possible to keep them separate. Here are the routes that were collapsed during analysis:

- Double Move → Wheel
- Slant In → Post
- Slant Out → Corner
- Z → In
- Comeback → Out
- Circle/Texas → Curl
- Skinny Post/Seam → Go

# Data preparation

After generating labels for each route the next challenge was developing ways to group the receivers on each play. The most straightforward option, using the label of each receiver active during the play, would lead to an impractically large number of route combinations to analyze and also hide a reality of the game -- on any given play QBs[1] are only surveying part of the field before throwing. To address these issues I used two approaches for grouping receivers:

1. **Pre-snap alignment-based grouping:** in this approach I considered all receivers who lined up on the same side of the field (either ‘top’ or ‘bottom’) as a group. During analysis I only used the group that contained the receiver who was ultimately targeted during the play (so if a WR from the ‘top’ of the formation was targeted I didn’t consider what the players on the ‘bottom’ of the formation were doing). This approach has the benefit of being simple to create and understand but it misses interactions from route combinations that align on opposed ends of the formation, like crossing routes. To better account for that type of scenario I also created a second grouping strategy:
2. **Receiver position at time of target-based grouping**: in this approach I consider all receivers who are within 20 yards of the targeted receiver when the ball arrives as being part of the same receiver group. As shown in the ‘Full Play’ plot in the Appendix, this approach allows receivers from opposite sides of the formation to be considered part of the same group. This approach was inspired by Chris Brown’s article on passing triangles [1], which shows that many plays are designed to get multiple receivers into similar parts of the field at the time of the pass, regardless of where they originally lined up. 20 yards may not be the optimal distance -- I manually checked a few dozen plays and 20 yards seemed reasonable but I would appreciate any feedback on that point.

[1] Except Tom Brady (🐐), of course.

Next, I computed several other variables to aid in the analysis:

1. I appended several fields from the ‘nflscrapR-data’ repo [4] to the plays data, such as win probability added (wpa), receiver_player_id (used to determine the target on passing plays), and yardline_100 (useful for computing whether the offense was in the red zone). I bridged the gap between the two types of player_ids by joining on name (completely unrelated fun fact: there were two Chris Thompsons in the NFL in 2017).
2. I also derived several variables:
    - **playSide** (player-play level variable): indicates whether the player was on the ‘top’ or ‘bottom’ of the given play, as defined by the relation to the center
    - **Has_target** (play level variable): indicates whether a receiver was targeted on the play. Pass plays that did not have a target were largely sacks or scrambles.
    - **Distance to_targeted_receiver** (player-play level variable): used this to create the ‘receiver position at time of  target’ grouping explained above.
    - **Team_formation_wpa** (team-formation level variable): the average WPA for pass plays with a targeted reciever for the given team with the given formation
    - **Wpa_minus_formation** (play level variable): the WPA of the play minus the ‘team_formation_wpa’. I use this as the main target variable in the analysis below. I computed this to try and isolate the impact of the receiver route combination from the team that happened to be running it. For example, if the Patriots successfully ran a lot of red zone fades and no one else did then using raw WPA might make it appear that red zone fades are the best route in the game, when really it’s only good when you have Tom Brady throwing to Gronk.
    - **In_redzone** (play level variable): self-explanatory indicator used during analysis.
    - **Personnel_offense_collapsed** (play level variable): I collapsed infrequent personnel groupings into a single ‘other’ category to aid analysis.
    - **Personnel_defense_collapsed** (play level variable):  I collapsed infrequent personnel groupings into a single ‘other’ category to aid analysis.

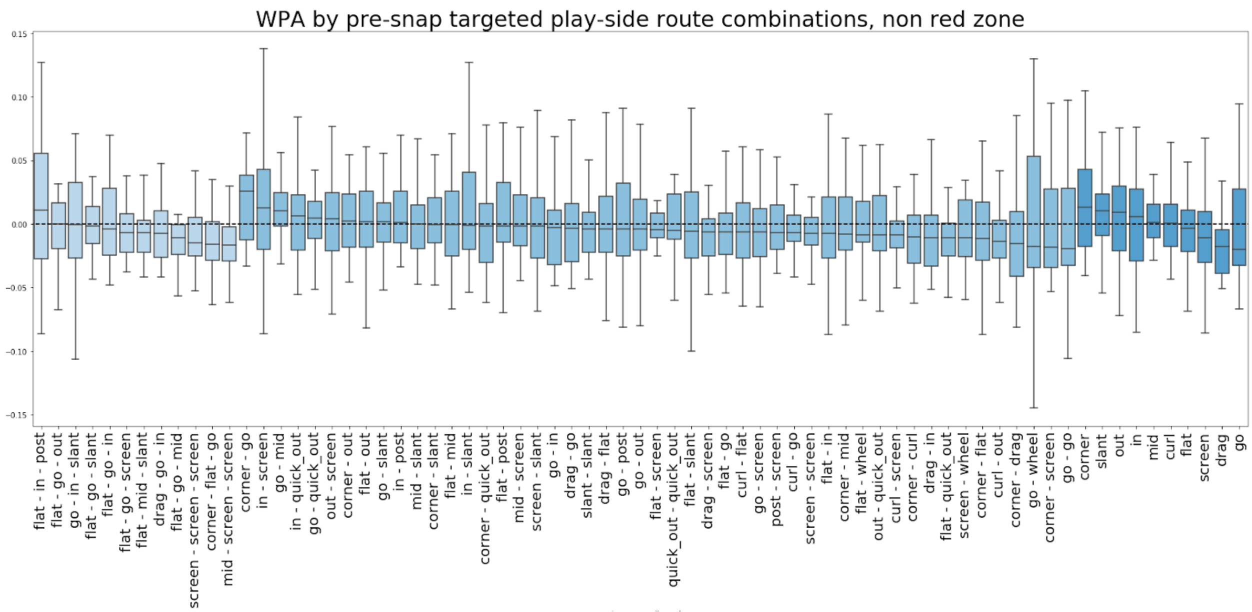
# Optimal receiver-route combination analysis

In this section I present the rank-ordered list of route combinations (via a box plot) along with OLS regression output that indicate the route combinations that are significant predictors (p <=.1) of success. I do this analysis for Red Zone (1-20 yards to TD) and Non-Red Zone plays in each of the receiver grouping methodologies I explained above, yielding 4 plots:

1. Pre-snap alignment-based grouping, Non Red Zone
2. Pre-snap alignment-based grouping, Red Zone
3. Receiver position at time of target-based grouping, Non Red Zone
4. Receiver position at time of target-based grouping, Red Zone

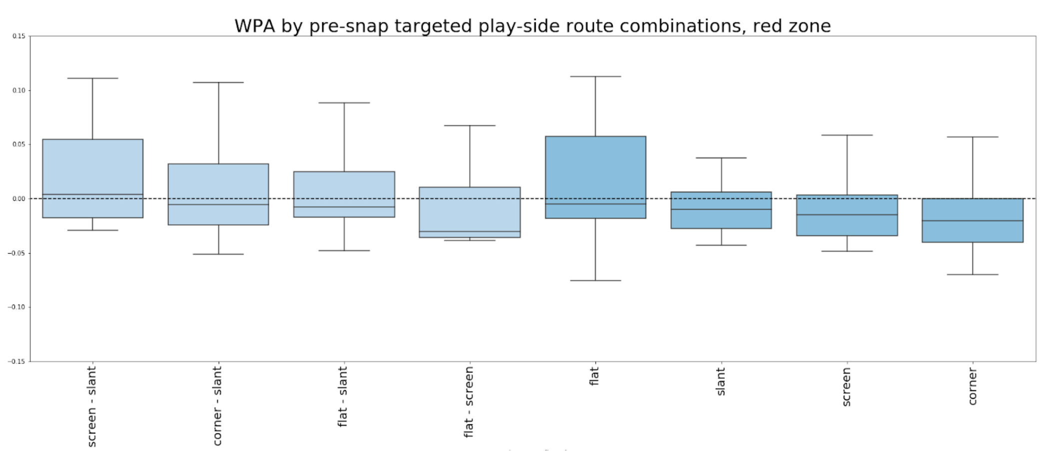
The box plots show the distribution of ‘WPA Minus Formation’ for all receiver group combinations that have at least X plays in the given section of the field -- for these charts the optimal value of X was scientifically determined to be 20 for the non-red zone and 10 for the red zone (kidding aside, I recognize this is a very small sample and would set the thresholds higher if more data were available). The plots are ordered by the number of receivers in the route group (which is also denoted with color) and then by the median ‘WPA Minus Formation’ -- so the best plays for each receiver-group size are on the left and the worst are on the right. The regressions used the observed receiver-route combinations and the defensive formation (both ‘dummified’ into categorical features).

### Pre-snap alignment-based groupings




| var | p_value | coef |
|---|---|---|
| drag | 0.0036 | -0.02 |
| corner | 0.0251 | 0.02 |
| corner - go | 0.0252 | 0.02 |
| go - mid | 0.0284 | 0.02 |
| flat - quick_out | 0.0364 | -0.02 |
| mid - screen - screen | 0.0473 | -0.01 |
| flat - in - post | 0.0728 | 0.02 |
| slant | 0.0899 | 0.01 |

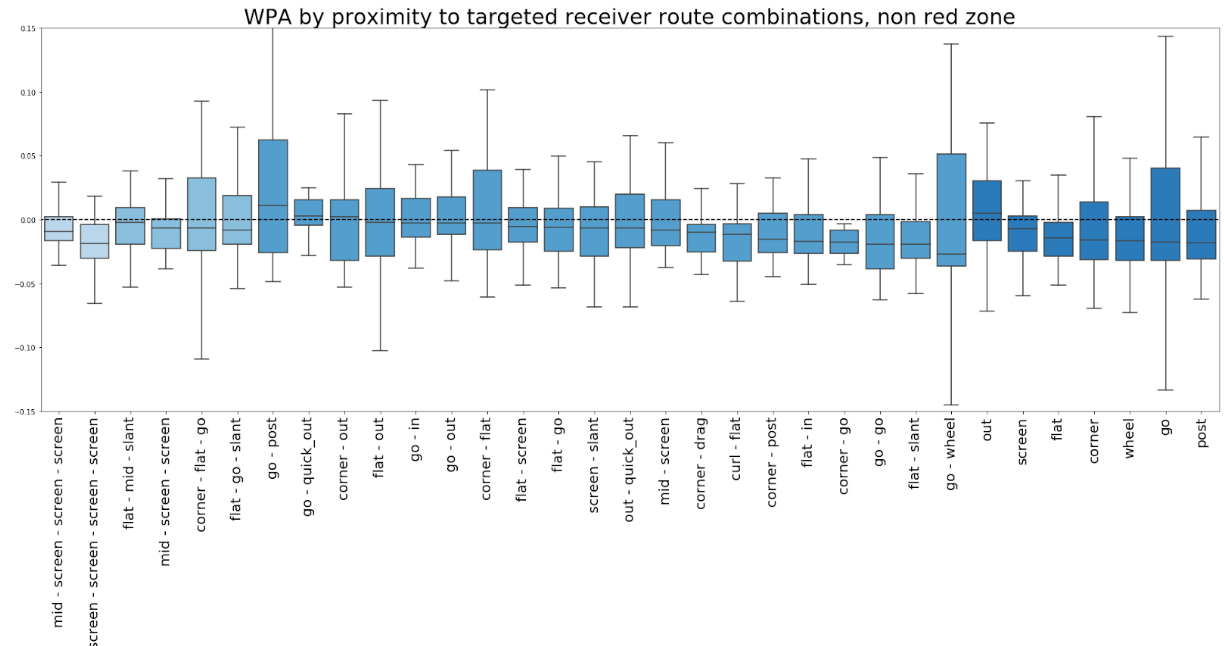
The top rated play in the three-receiver group is Steve Spurrier’s favorite -- the “Mills Play” [2], denoted here as ‘Flat - In - Post’. This play famously stresses multiple parts of the field and is a worthy holder of the top ranked spot. ‘Corner - Go’, a route combo that stresses the last line of defense, tops the two-receiver group, while ‘Drag’ is the least effective route. The regression produces similar output while also calling out ‘Corner’ as an effective route. The main story, unfortunately, is that the receiver-route effects are muted -- no combos show large effects.



| var | p_value | coef |
|---|---|---|
| dime | 0.0438 | 0.03 |

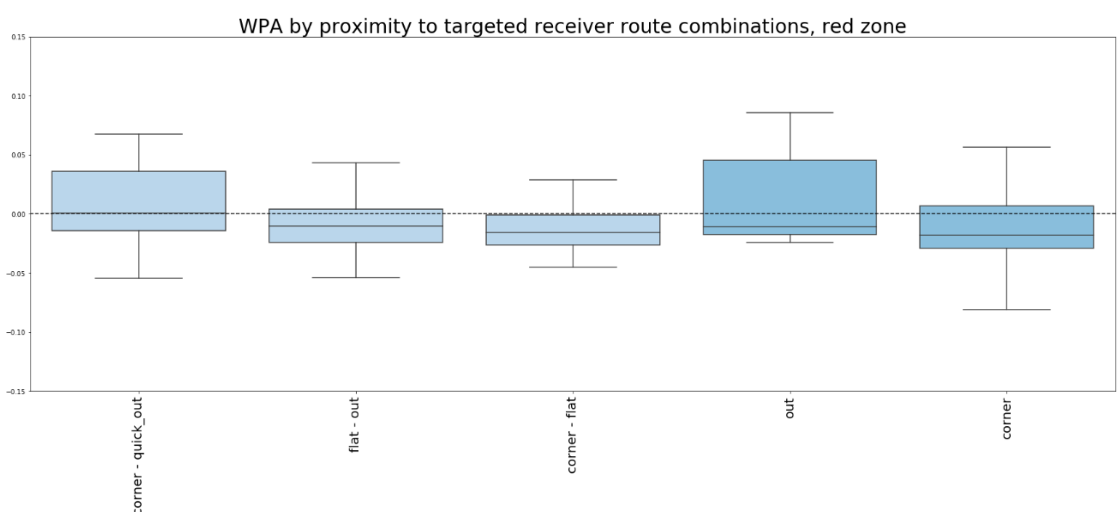
The red zone route combinations show a similarly muted pattern, though there is some evidence that when calling one-man routes coaches should attack the flat instead of the corner; better yet, they should throw a two-man route with a slant. The regression output holds a lesson for defensive coaches -- ‘don’t play dime in the red zone’ probably won’t net me any points for innovation but it’s good to see common sense validated.

## Receiver position at time of target-based groupings




| var | p_value | coef |
|---|---|---|
| go - post | 0.0002 | 0.03 |
| out | 0.0843 | 0.01 |

Grouping receivers by their position at the time of the ball’s arrival causes some of the route combinations shown above to fall out of the sample and others to come in their place. The newly arrived four-man screen routes don’t grade out well but the ‘Go - Post’ combo looks good.



Like the ‘pre-snap’ version above, the ‘position at target’ red zone chart doesn’t love throwing the corner as a one man route in the red zone, showing that out routes tend to find better success. No route combinations were significant in the regression for this game situation + route grouping.

Overall this analysis of receiver-route combinations did not unearth any startling new insights but it did validate that the Head Ball Coach (and this armchair quarterback) were onto something -- airing it out with receiver route combinations like the ‘Mills Play’, ‘Corner - Go’, and ‘Go - Post’ helps teams win[2].

# Future work

With more time I’d want to work on a few things: (1) develop alternate metrics of receiving success, such as separation at the time of the pass and/or pass arrival (ideally we would know the play’s planned pass release time, but this would be a reasonable proxy) that can be calculated for each receiver on each play; (2) develop a classifier for identifying man vs. zone defense, and then use that when evaluating each receiver’s success; and (3) re-train RouteNet with cleaner (and more) labels.

# Conclusion

In this paper I introduced RouteNet, a convolutional neural network that classifies routes, and used RouteNet’s classifications to compute some basic statistics on the effectiveness of receiver-route combinations. I did not uncover any startling insights but am confident this type of automated route recognition will soon become commonplace inside NFL front offices (if it is not already) and will quickly yield even higher quality classifications when trained with additional examples.

The player-level tracking data provided for the Big Data Bowl is incredible -- it is accurate, at a high time resolution, and well organized. I enjoyed playing around with it and hope I have the opportunity to do so again.

Despite the data’s strengths, though, working through this exercise hammered home a fundamental problem of football analytics that I don’t yet know how to tackle[3]: there are only about 150 plays in a game so data will always be regrettably scarce. To push football analytics forward we will need to continue researching with an enthusiasm unknown to mankind and develop novel ways to extract the most meaning out of the few trials we get each year.

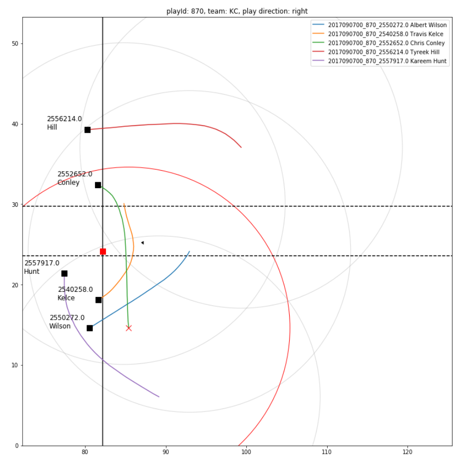
---

[2] Especially when the impact of plays without a target receiver, like sacks, which disproportionately affect deep passing plays, are ignored due to not having a good method for linking their impact to individual route combinations.

[3] Sorry -- I made it this far without a terrible pun and just couldn’t resist any longer.

# APPENDIX

**'Full play' plot**: these were used for initial manual route classification and for determining the optimal distance from the targeted receiver to use when creating the 'post-snap' receiver groups. The red 'X' shows the location of the reception and the large red circle is drawn with a radius of 20 yards from that spot. This example shows the benefit of creating receiver groups based on the location of the reception -- Travis Kelce lined up on the opposite side of the formation of the target (and so would not be included in pre-snap play-side analysis) but his drag route helped create space for Chris Conley, the receiver targeted on the play..



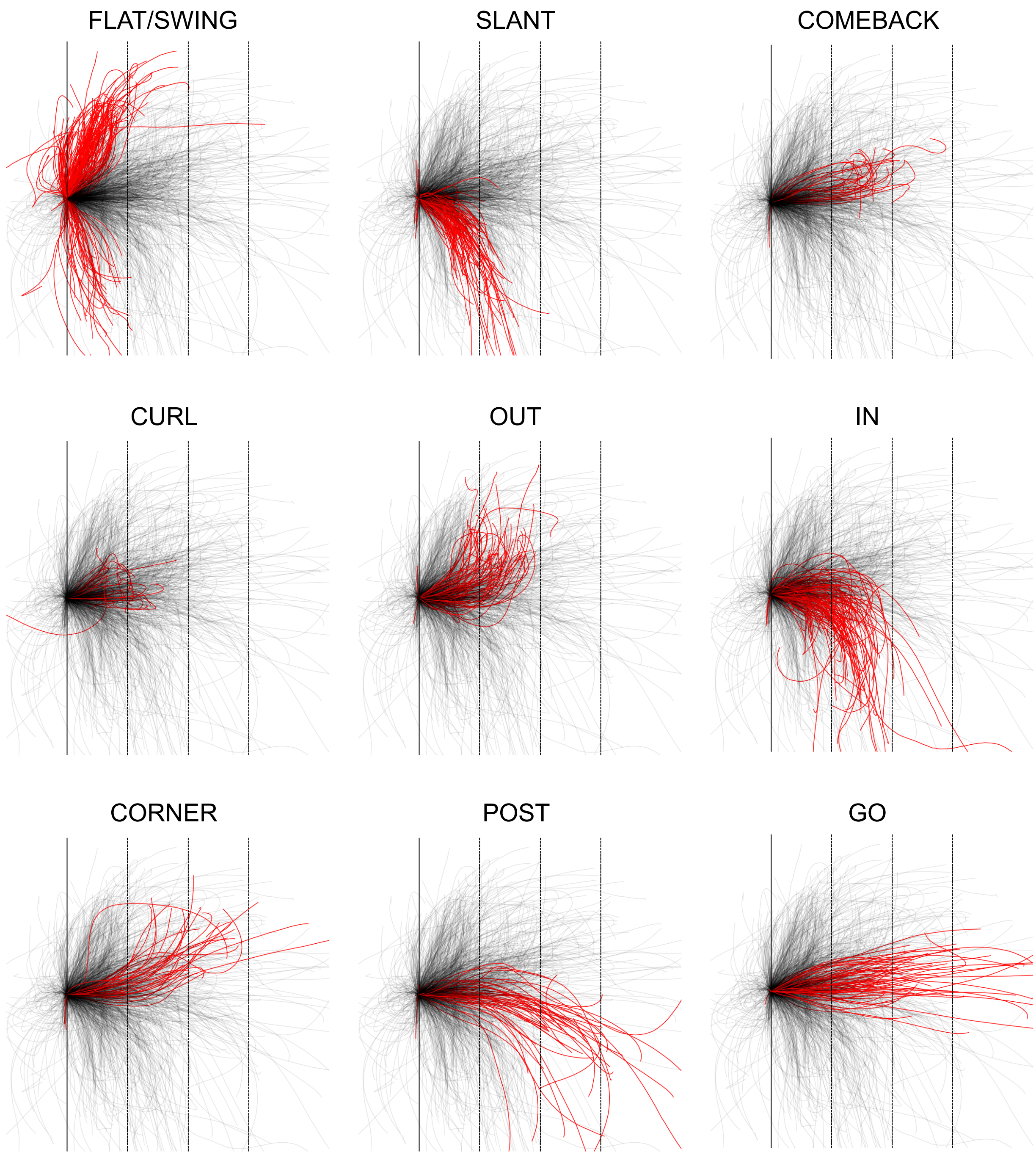

### RouteNet Confusion Matrix

| | corner | curl | drag | flat | go | in | mid | out | post | quick_out | screen | slant | wheel | TOTAL LABELS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| corner | 38 | 1 | 0 | 0 | 11 | 0 | 0 | 4 | 0 | 0 | 0 | 0 | 6 | 60 |
| curl | 0 | 28 | 0 | 2 | 8 | 10 | 8 | 12 | 1 | 6 | 6 | 2 | 1 | 84 |
| drag | 0 | 0 | 52 | 3 | 0 | 0 | 0 | 0 | 0 | 6 | 0 | 8 | 0 | 69 |
| flat | 2 | 2 | 10 | 135 | 0 | 0 | 5 | 0 | 0 | 8 | 14 | 1 | 1 | 178 |
| go | 20 | 3 | 0 | 0 | 147 | 3 | 5 | 3 | 9 | 0 | 0 | 3 | 4 | 197 |
| in | 0 | 3 | 1 | 0 | 7 | 85 | 1 | 1 | 9 | 1 | 0 | 10 | 1 | 119 |
| mid | 6 | 4 | 1 | 5 | 8 | 0 | 67 | 0 | 0 | 8 | 9 | 3 | 2 | 113 |
| out | 10 | 11 | 4 | 0 | 0 | 2 | 0 | 40 | 1 | 2 | 0 | 0 | 1 | 71 |
| post | 0 | 0 | 0 | 0 | 9 | 14 | 0 | 0 | 37 | 0 | 0 | 7 | 1 | 68 |
| quick_out | 0 | 5 | 4 | 1 | 0 | 1 | 3 | 3 | 0 | 30 | 0 | 0 | 0 | 47 |
| screen | 0 | 0 | 0 | 11 | 0 | 0 | 6 | 0 | 0 | 0 | 126 | 0 | 0 | 143 |
| slant | 0 | 1 | 3 | 0 | 4 | 13 | 8 | 0 | 9 | 0 | 0 | 63 | 0 | 101 |
| wheel | 4 | 1 | 2 | 6 | 6 | 1 | 0 | 1 | 2 | 1 | 0 | 3 | 19 | 46 |
| **TOTAL PREDS** | 80 | 59 | 77 | 163 | 200 | 129 | 103 | 64 | 68 | 62 | 155 | 100 | 36 | |

| | corner | curl | drag | flat | go | in | mid | out | post | quick_out | screen | slant | wheel | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| MATCHES | 38 | 28 | 52 | 135 | 147 | 85 | 67 | 40 | 37 | 30 | 126 | 63 | 19 | 867 | |
| PRECISION | 48% | 47% | 68% | 83% | 74% | 66% | 65% | 63% | 54% | 48% | 81% | 63% | 53% | **69%** | * weighted average |
| TOTAL LABELS | 60 | 84 | 69 | 178 | 197 | 119 | 113 | 71 | 68 | 47 | 143 | 101 | 46 | 1296 | |
| RECALL | 63% | 33% | 75% | 76% | 75% | 71% | 59% | 56% | 54% | 64% | 88% | 62% | 41% | **70%** | * weighted average |

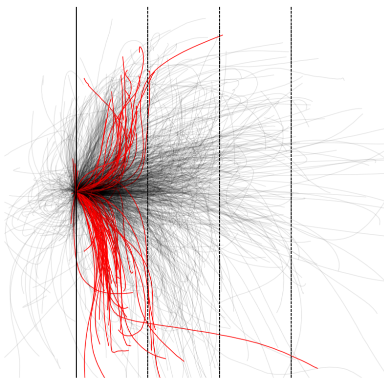
## RouteNet class validation plots (pt 1 of 2)

Each plot below contains all routes run by an eligible receiver across four games in week three (read: enough plays to see the pattern but not so many as to be unreadable). The red routes are those classified by RouteNet as matching the above label. All location traces are centered to the same point in the plot, regardless of their location to the line of scrimmage in each play.

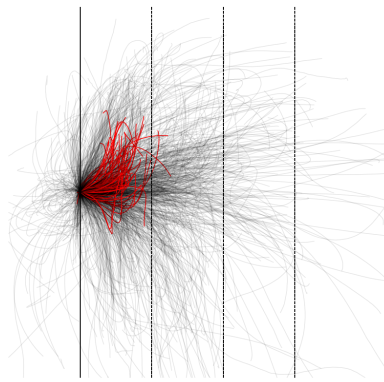


# RouteNet class validation plots (pt 2 of 2)

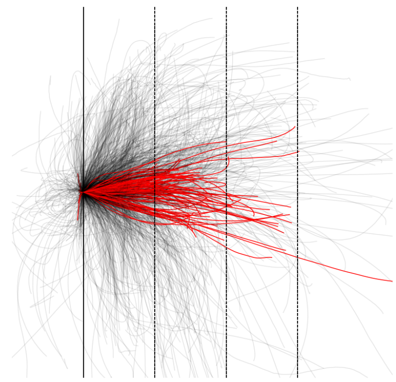
DRAG



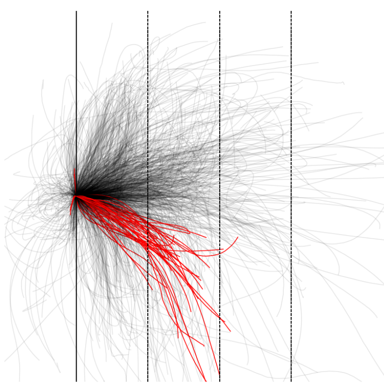
QUICK OUT



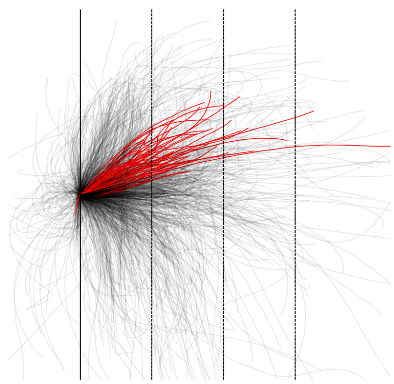
SEAM/SKINNY POST



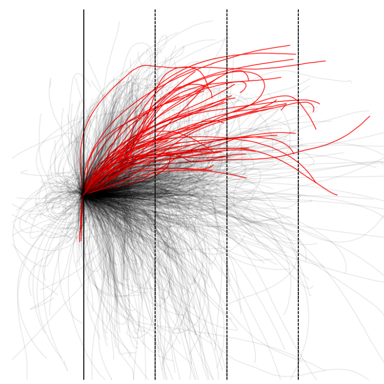
“SLANT IN”



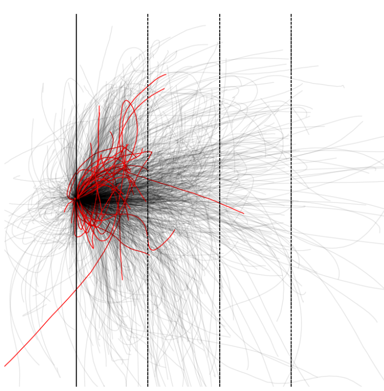
“SLANT OUT”



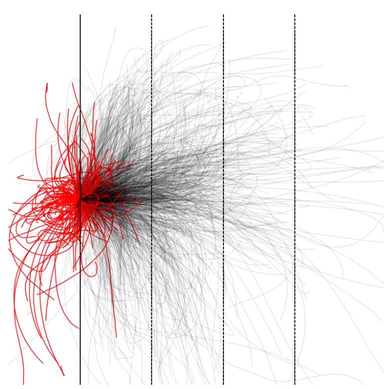
WHEEL



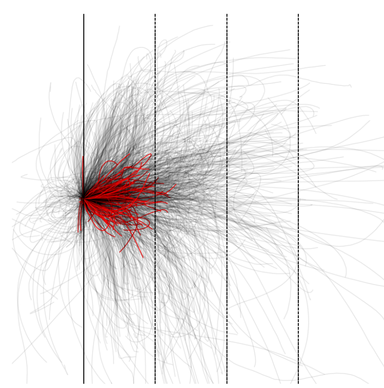
CIRCLE / TEXAS



SCREEN/BLOCK



“MID”



## References

[1] Brown, ‘Snag, stick, and the importance of triangles (yes, triangles) in the passing game’. http://smartfootball.com/passing/snag-stick-and-the-importance-of-triangles-yes-triangles-in-the-passing-game

[2] Brown, ‘The Science of the Post: Going Deep with “Mills”’. http://smartfootball.com/offense/the-science-of-the-post-going-deep-with-mills

[3] Hochstedler and Gagnon, ‘American Football Route Identification Using Supervised Machine Learning’. http://www.sloansportsconference.com/wp-content/uploads/2017/02/1542.pdf

[4] Yurko, nflscrapR-data. https://github.com/ryurko/nflscrapR-data

[5] Yurko, Ventura, and Horowitz. ‘nflWAR: A Reproducible Method for Offensive Player Evaluation in Football’. https://arxiv.org/abs/1802.00998