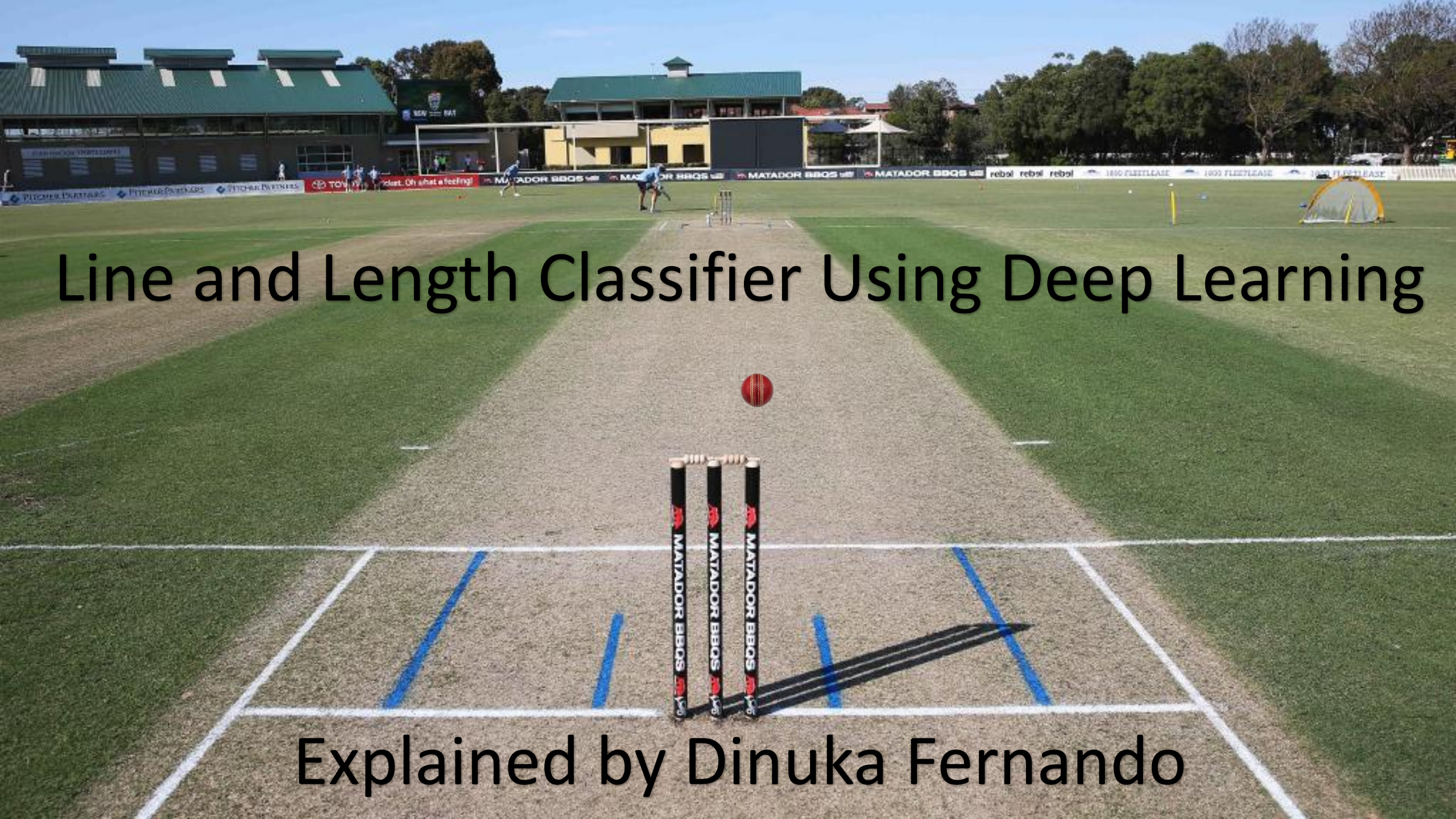
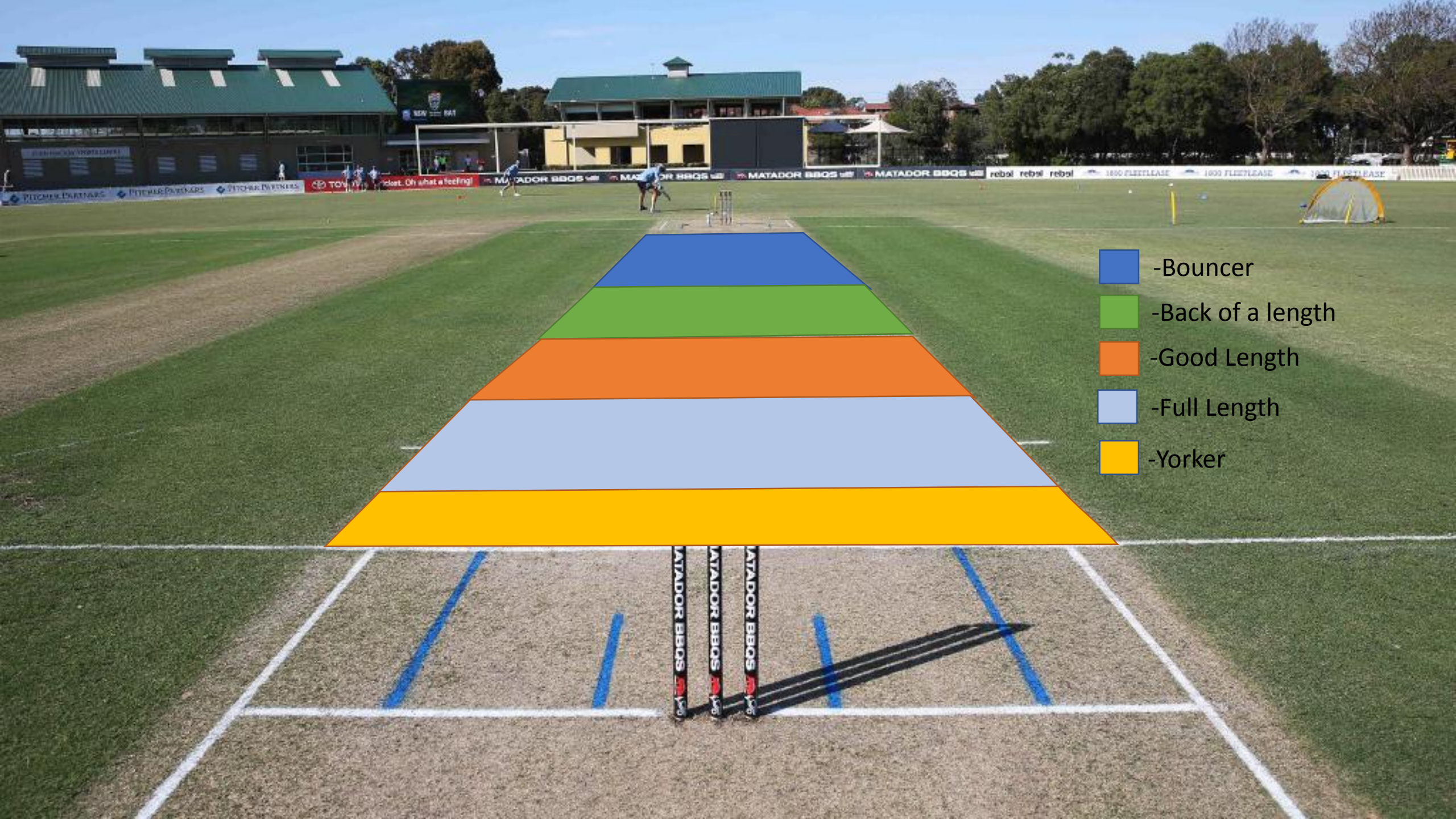
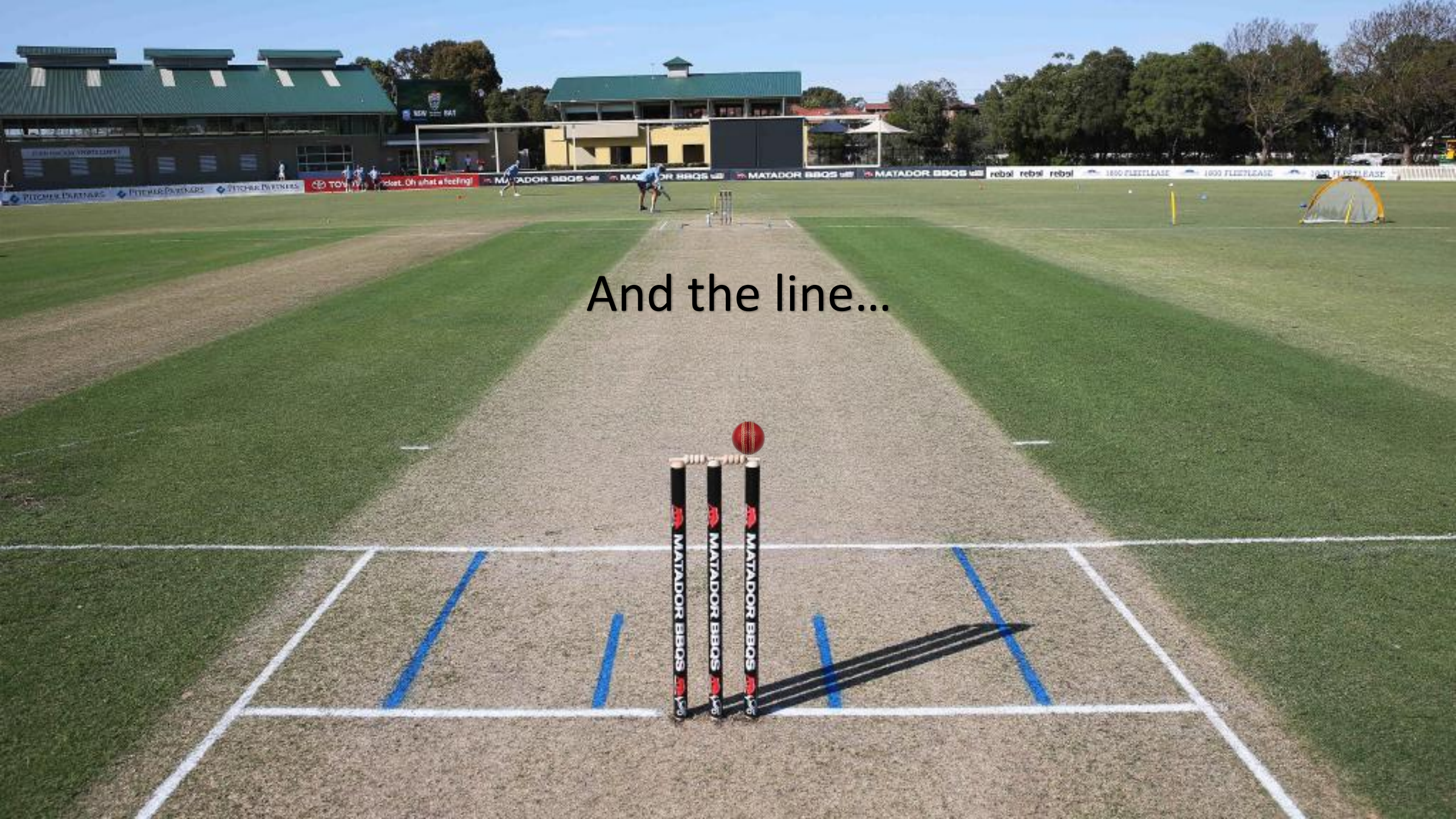

Line and Length Classifier Using Deep Learning
Explained by Dinuka Fernando

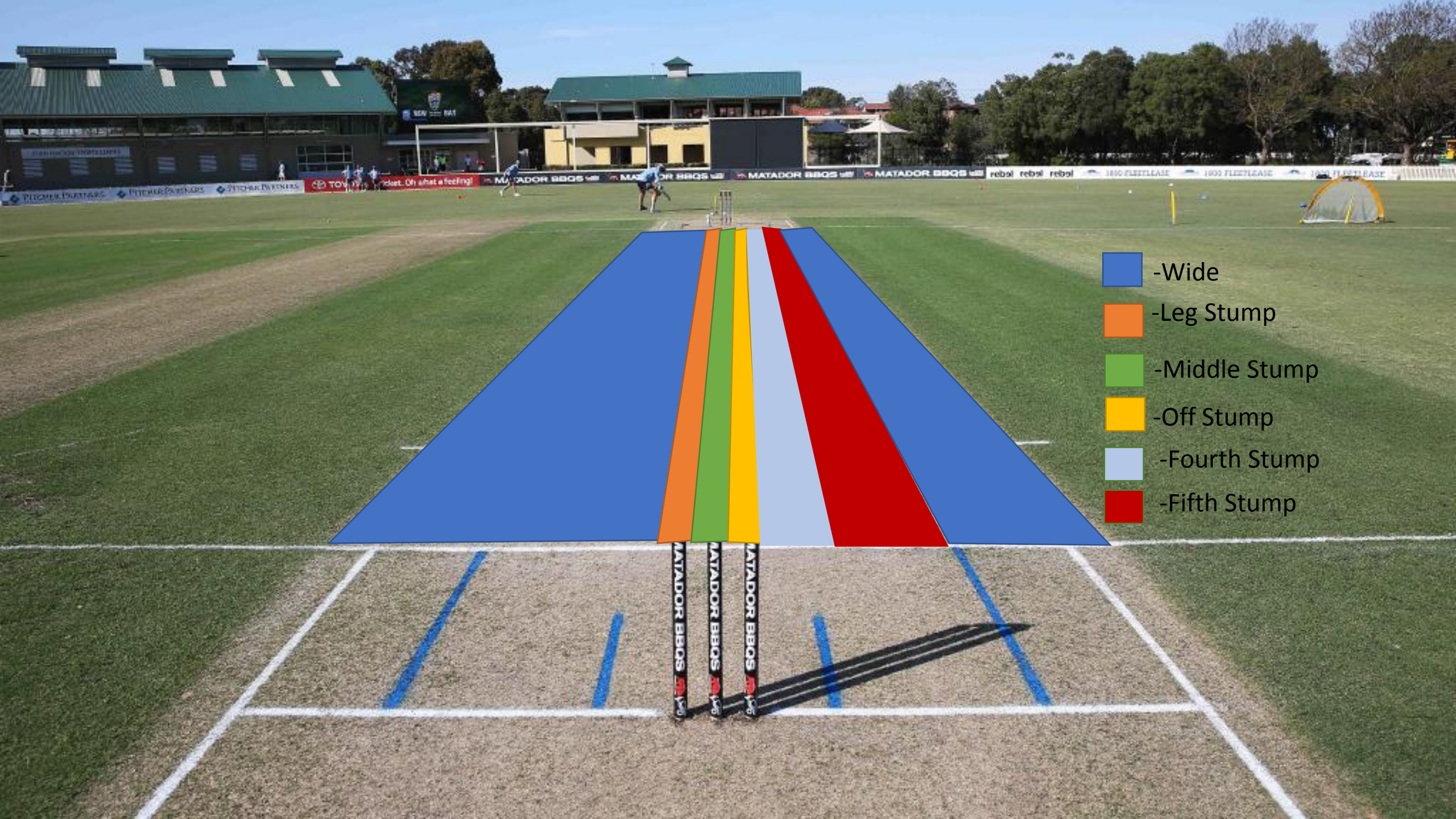
Imagine if you are a right handed batsman facing a fast bowler.

[1][2]

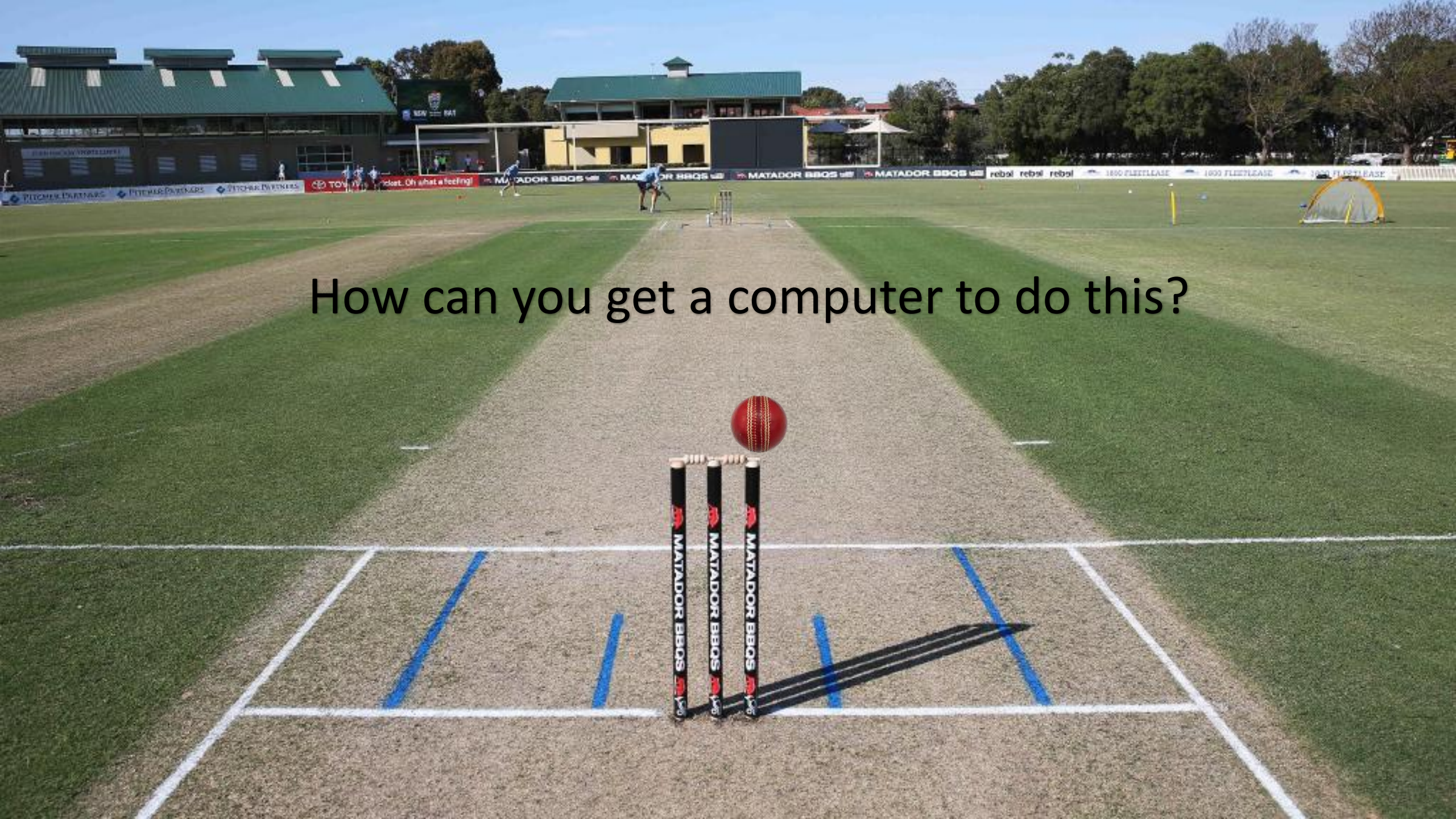

You need to judge the length…
MATADOR BBQS
rebel


-Bouncer
-Back of a length
-Good Length
-Full Length
-Yorker


And the line…
MATADOR BBQS
rebel

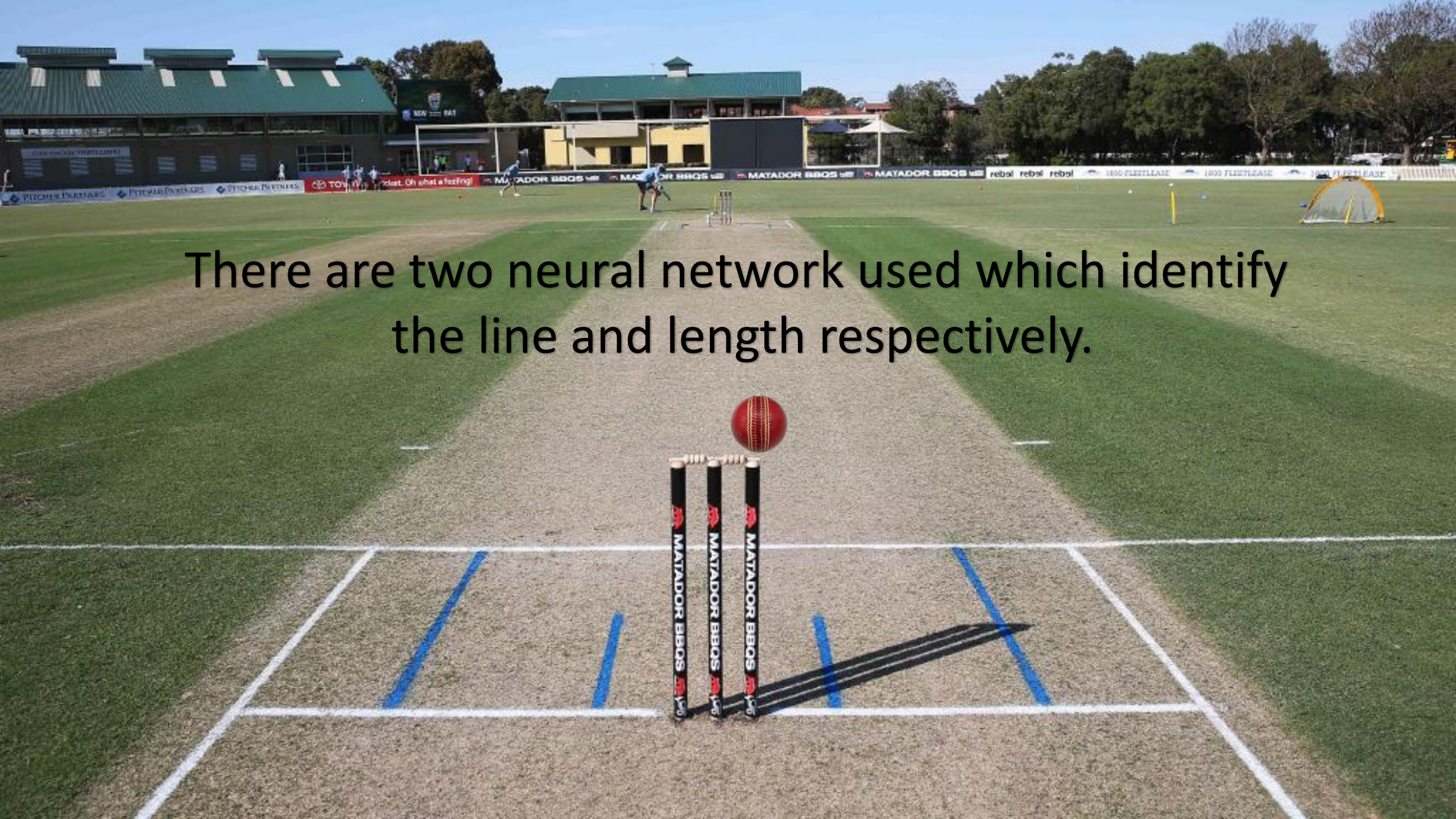

-Wide
-Leg Stump
-Middle Stump
-Off Stump
-Fourth Stump
-Fifth Stump
MATADOR BBQS
rebel


This is a full length delivery pitching at off stump.
MATADOR BBQS
MATADOR BBQS
MATADOR BBQS
rebel rebel rebel


How can you get a computer to do this?


This process involves supervised learning.
MATADOR BBQS
MATADOR BBQS
MATADOR BBQS

There are two neural network used which identify the line and length respectively.



I created 100 images of different deliveries using PowerPoint.

The pictures were labelled using a csv file.

80 images were used for the training set.

In other words, the computer tries to recognize the 80 images.

Eg.

Top image- Good length.

Middle Image-Full length

Bottom Image-Back of a length

20 images were used for the validation set.

In other words, the computer tries to classify 20 images which it has not seen before.

Eg.

Top image- Full length.

Middle Image-Bouncer

Bottom Image-Yorker

The neural networks are created using the fastAi library.
The dataset has been trained using a pretrained model (resnet50).

Default settings are used as a starting point.

```
learnLength = cnn_learner(lengthDataBunch, models.resnet50, metrics=accuracy)
```

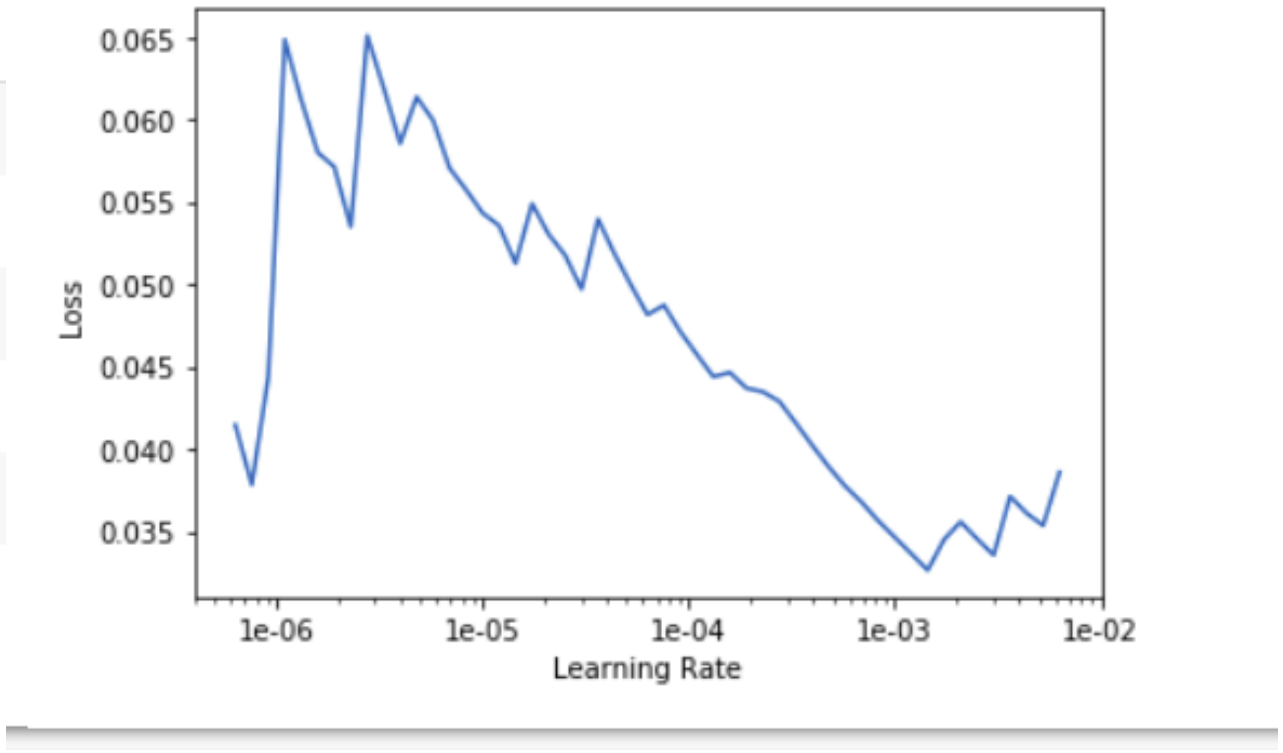
This pretrained model can already classify multiple objects. Only the weights near the last few layers are tweaked accordingly.

There are AI researchers who use this technique to achieve higher accuracy whilst reducing time.

[4]

The initial results for detecting length are shown below.

epoch	train_loss	valid_loss	accuracy	time
0	2.487618	2.334050	0.150000	00:03
1	1.289890	2.261957	0.250000	00:03
2	0.887085	1.554783	0.350000	00:03
3	0.646340	1.387026	0.350000	00:03
4	0.509642	1.385116	0.350000	00:03



In the accuracy column 1 represent 100%.

Parameters are tweaked in order to improve results.

```
learnLength.fit_one_cycle(10, max_lr=slice(1e-4,1.5e-4) , wd = 0.3)
```

Set a condition on the weights to reduce underfitting.

epoch	train_loss	valid_loss	accuracy	time
0	0.111469	0.798398	0.700000	00:03
1	0.072062	0.649933	0.800000	00:03
2	0.122023	0.811473	0.800000	00:03
3	0.090593	1.311500	0.750000	00:03
4	0.082717	1.078450	0.900000	00:03
5	0.082713	0.880208	0.900000	00:03
6	0.069507	0.878031	0.800000	00:03
7	0.079710	1.026887	0.800000	00:03
8	0.071355	1.045996	0.850000	00:03
9	0.069311	1.055478	0.900000	00:03

Tweak the learning rates of all layers in the CNN from 1e-4 to 1.5e04.

Learning rates refer to how quickly the computer learns.
High learning rate-Miss key details
Low learning rate- Too slow

Ideally you want train_loss <valid_loss. This is a good sign.

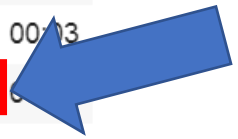
The computer is able to identify the correct length 90% of the time.

[3][4]

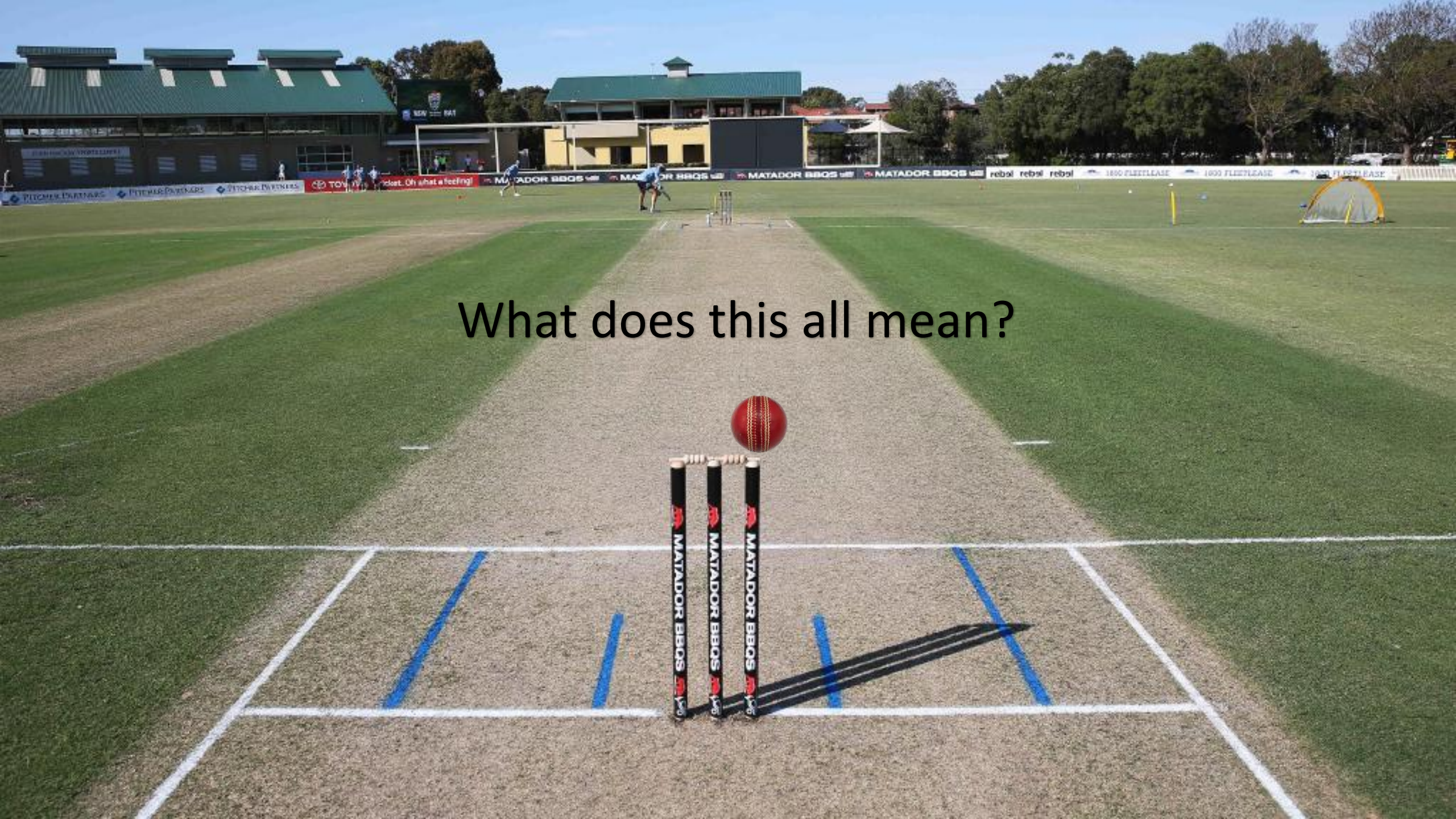
Here are the final results for the line classification model.

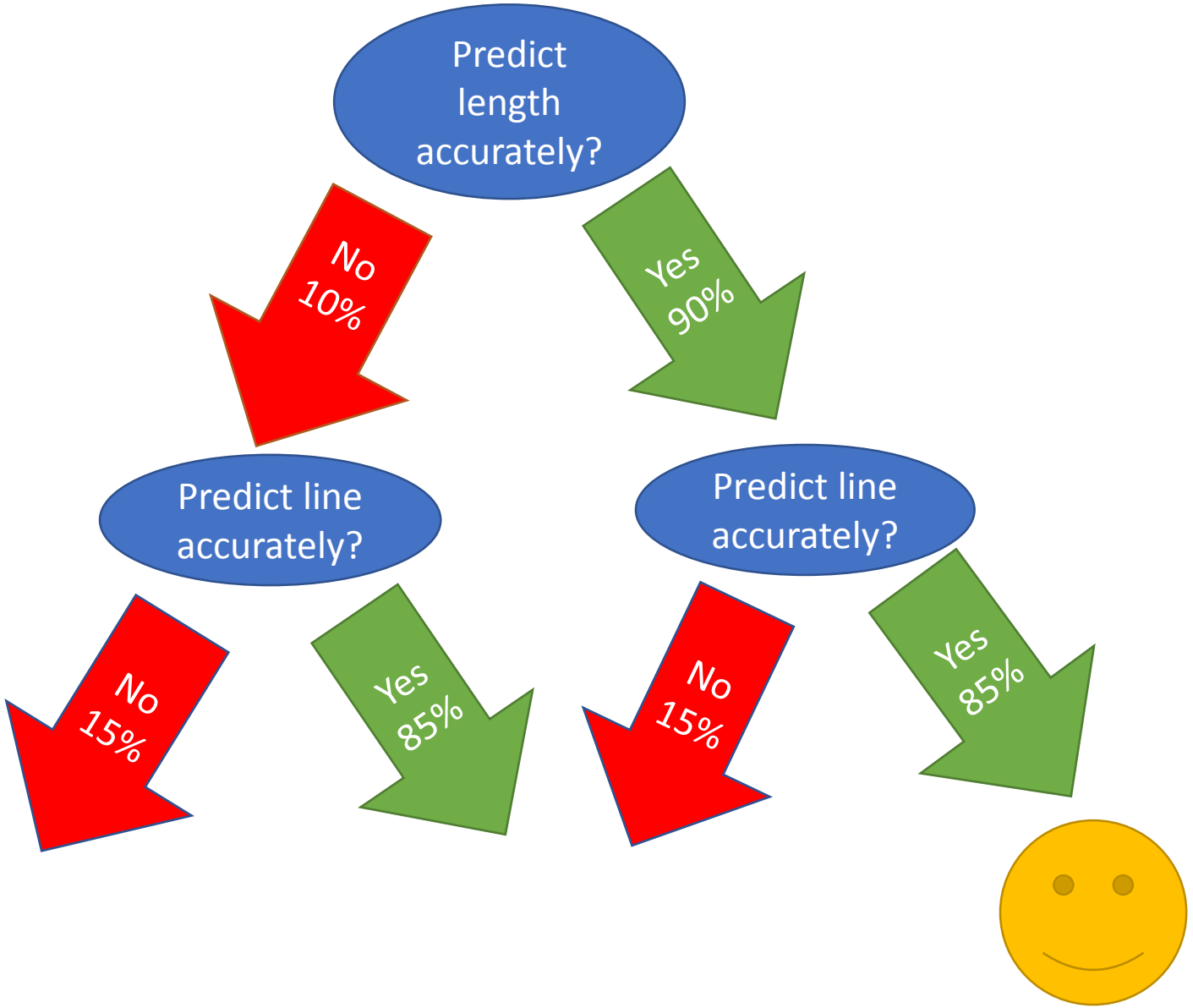
```
learnLength.fit_one_cycle(10, max_lr=slice(1e-4,1.6e-4) , wd = 0.5)
```

epoch	train_loss	valid_loss	accuracy	time
0	0.047111	0.307772	0.850000	00:03
1	0.038165	0.295550	0.850000	00:03
2	0.042266	0.966980	0.750000	00:03
3	0.055058	0.826749	0.800000	00:03
4	0.055858	0.531523	0.850000	00:03
5	0.072878	0.943986	0.800000	00:03
6	0.088196	0.885688	0.850000	00:03
7	0.099183	0.813027	0.800000	00:03
8	0.105549	0.768120	0.800000	00:03
9	0.097884	0.791694	0.800000	00:03



The computer is able to identify the line correctly 85% of the time.


What does this all mean?
MATADOR BBQS
rebel



Using basic math, the probability that the neural networks will accurately is simply:

(0.9 * 0.85) * 100
= 76.5%

Still a work in progress.

Image References:

[1] Whack Sports, *WHACK-test-4-pce-156g-red-cricket-ball*. 2020.

[2] The Daily Telegraph, *The wicket area is rolled for an hour a day to create firmness on the pitch. Phil Hillyard*. 2020.

AI References:

[3] J. Howard, *Lesson 1: Image classification*. 2020.

[4] “fastai", *Docs.fast.ai*, 2020. [Online]. Available: https://docs.fast.ai/. [Accessed: 23- Jan- 2020].

Thank you for watching.