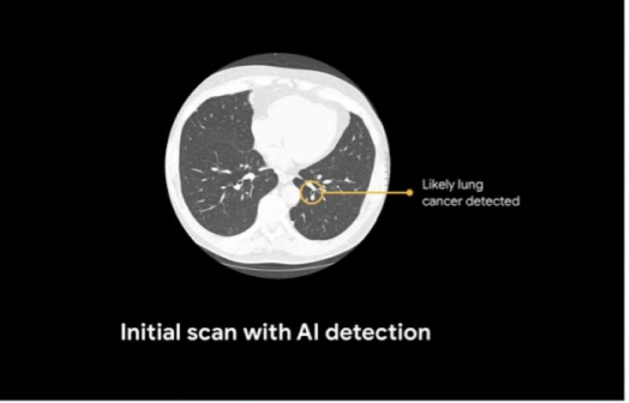
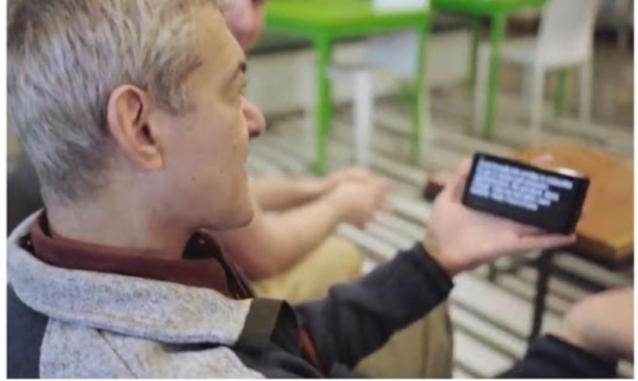
May the Tensors Flow!

Laurence Moroney
@lmoroney

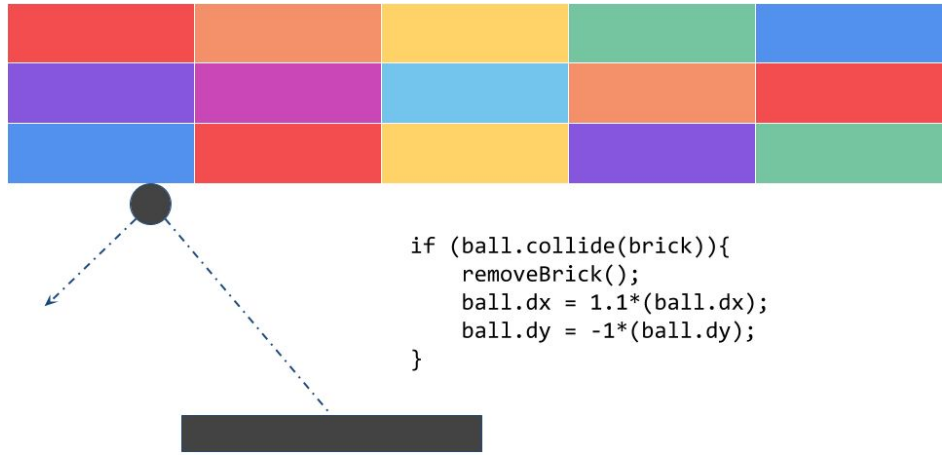

Likely lung
cancer detected
Initial scan with AI detection



TensorFlow


8-7=1
8-7=1
7+4=1
8-7=1
8-7=1

Explicit Coding



Defining rules that determine behavior of a program

Everything is pre-calculated and pre-determined by the programmer

Scenarios are limited by program complexity

The Traditional Programming Paradigm

Consider Activity Detection

```
if(speed<4){
    status=WALKING;
}
```

Consider Activity Detection

```
if(speed<4){
    status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

Consider Activity Detection

```
if(speed<4){
    status=WALKING;
}
```

```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

Consider Activity Detection

```
if(speed<4){
    status=WALKING;
}
```

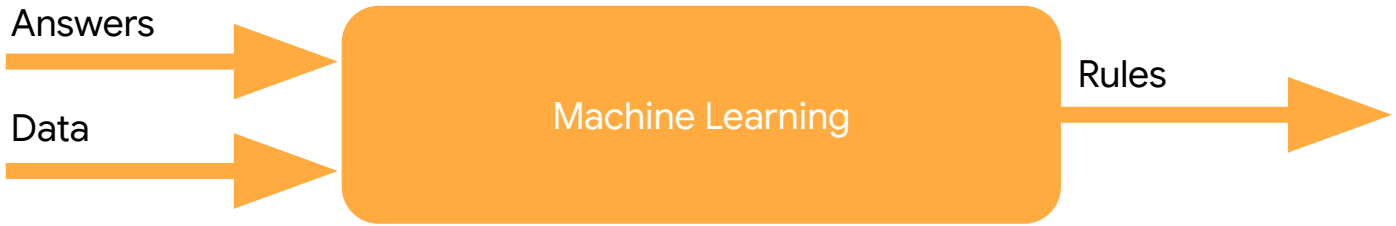
```
if(speed<4){
    status=WALKING;
} else {
    status=RUNNING;
}
```

```
if(speed<4){
    status=WALKING;
} else if(speed<12){
    status=RUNNING;
} else {
    status=BIKING;
}
```

```
// ???
```

The Traditional Programming Paradigm

The Machine Learning Paradigm



Activity Detection with Machine Learning

```
0101001010100101010
1001010101001011101
0100101010010101001
0101001010100101010
```

Label = WALKING

```
1010100101001010101
0101010010010010001
0010011111010101111
1010100100111101011
```

Label = RUNNING

```
1001010011111010101
1101010111010101110
1010101111010101011
1111110001111010101
```

Label = BIKING

```
1111111111010011101
0011111010111110101
0101110101010101110
1010101010100111110
```

Label = GOLFING

The Machine Learning Paradigm

```
01010010101100101010
10010101010011011101
01001010100101011001
01010010101001010010
```

Label = WALKING

```
10101001010010101010
01010100100100010001
00100111110101011111
10101001001111010111
```

Label = RUNNING

```
10010100111110101010
11010101110101011110
10101011110101010111
11111100011110101010
```

Label = BIKING

```
11111111110100111010
00111111010111110101
01011101010101011110
10101010100111110
```

Label = GOLFING

The Machine Learning Paradigm

```
0101001010100101010
1001010101001011101
0100101010010101001
0101001010100101010
```

Label = WALKING

```
1010100101001010101
0101010010010010001
0010011111010101111
1010100100111101011
```

Label = RUNNING

```
1001010011111010101
1101010111010101110
1010101111010101011
1111110001111010101
```

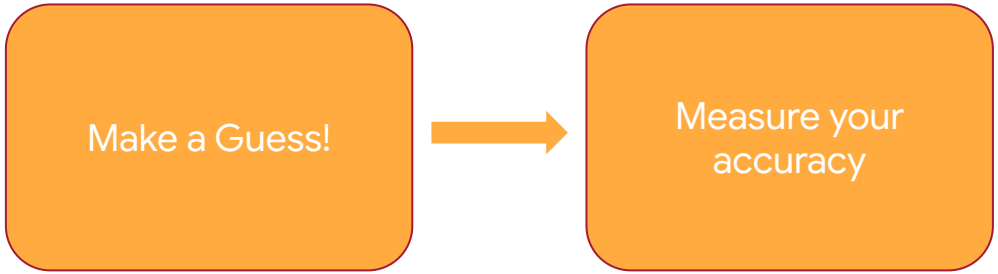
Label = BIKING

```
1111111111010011101
0011111010111110101
0101110101010101110
1010101010100111110
```

Label = GOLFING

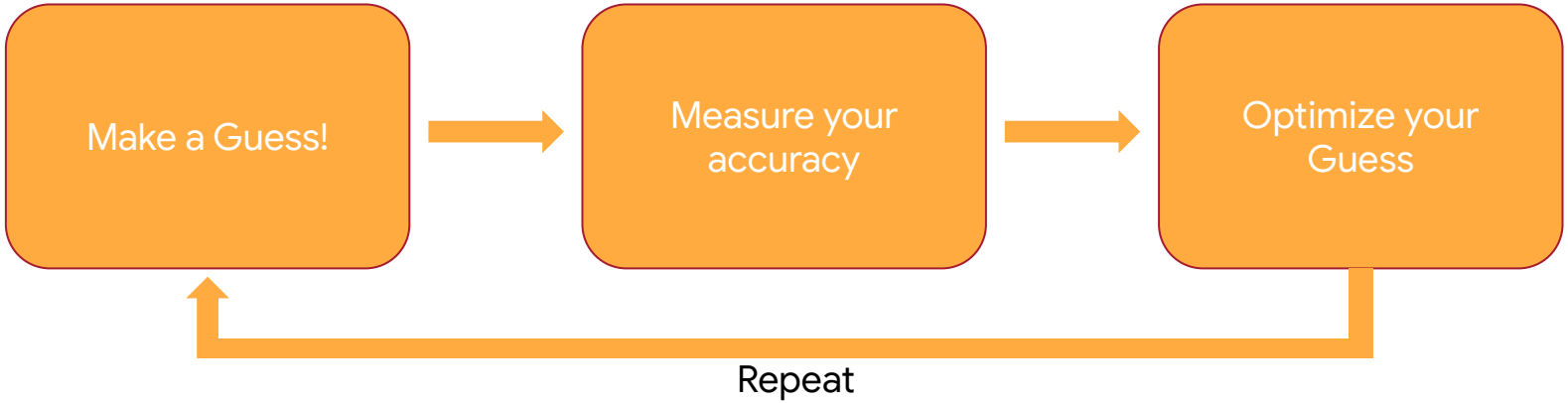
The Machine Learning Paradigm

The Machine Learning Paradigm

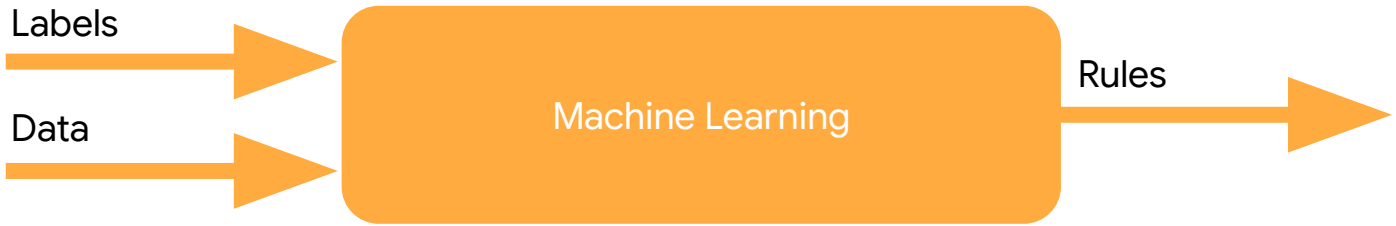


The Machine Learning Paradigm

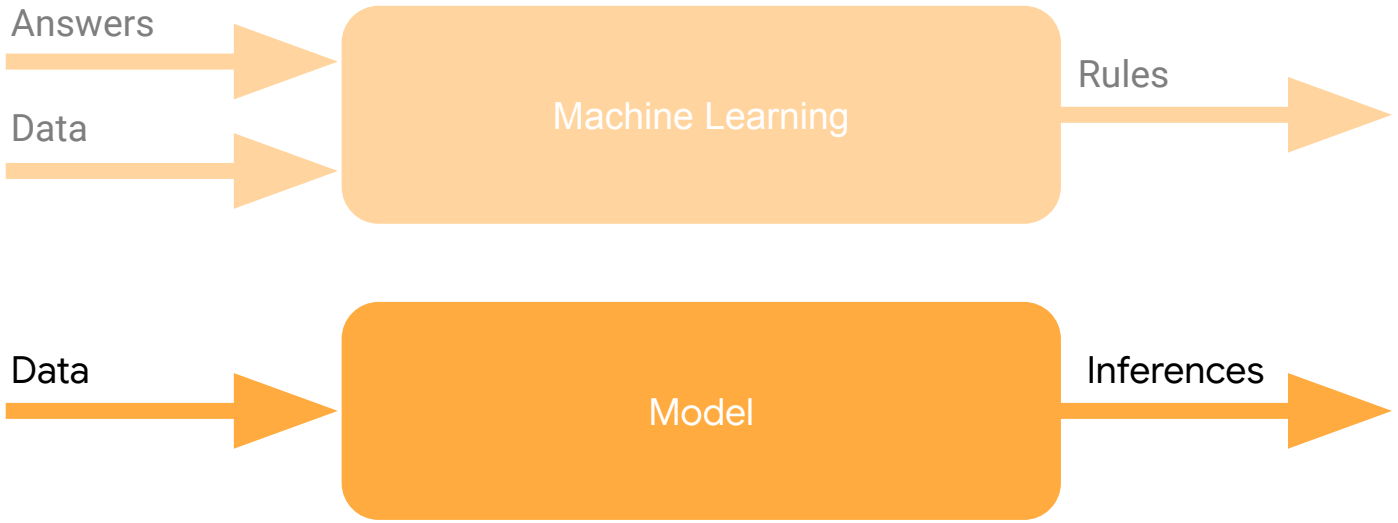
The Machine Learning Paradigm



The Machine Learning Paradigm



The Machine Learning Paradigm



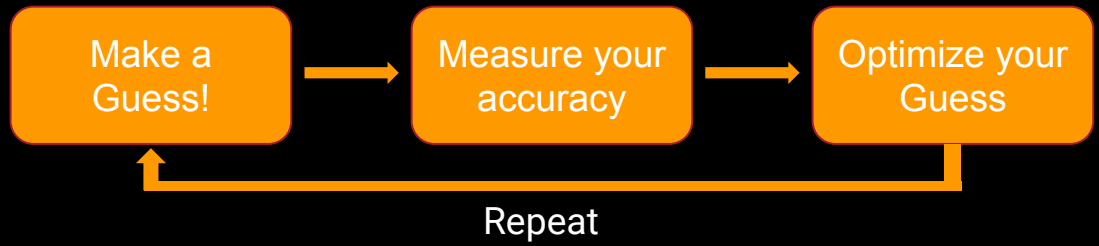
X = -1, 0, 1, 2, 3, 4

Y = -3, -1, 1, 3, 5, 7

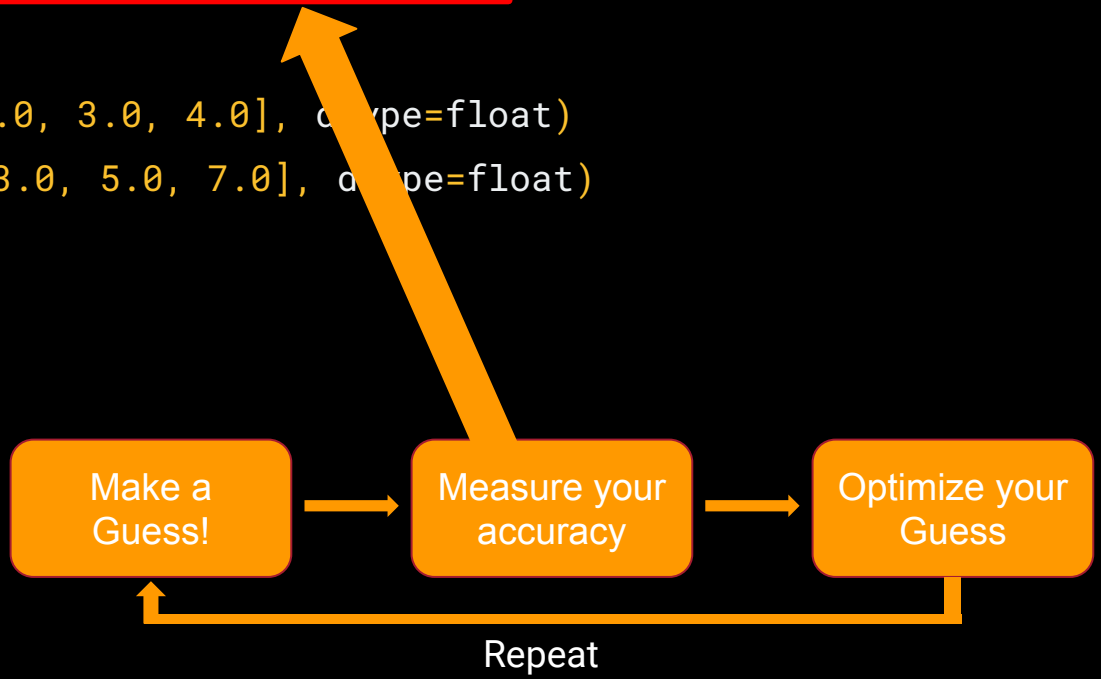
```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

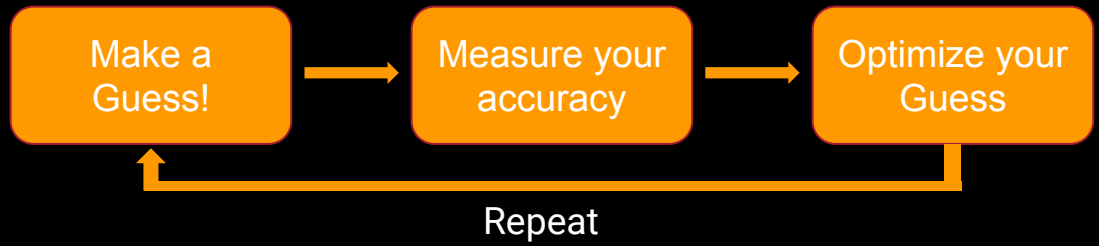


```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```



```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

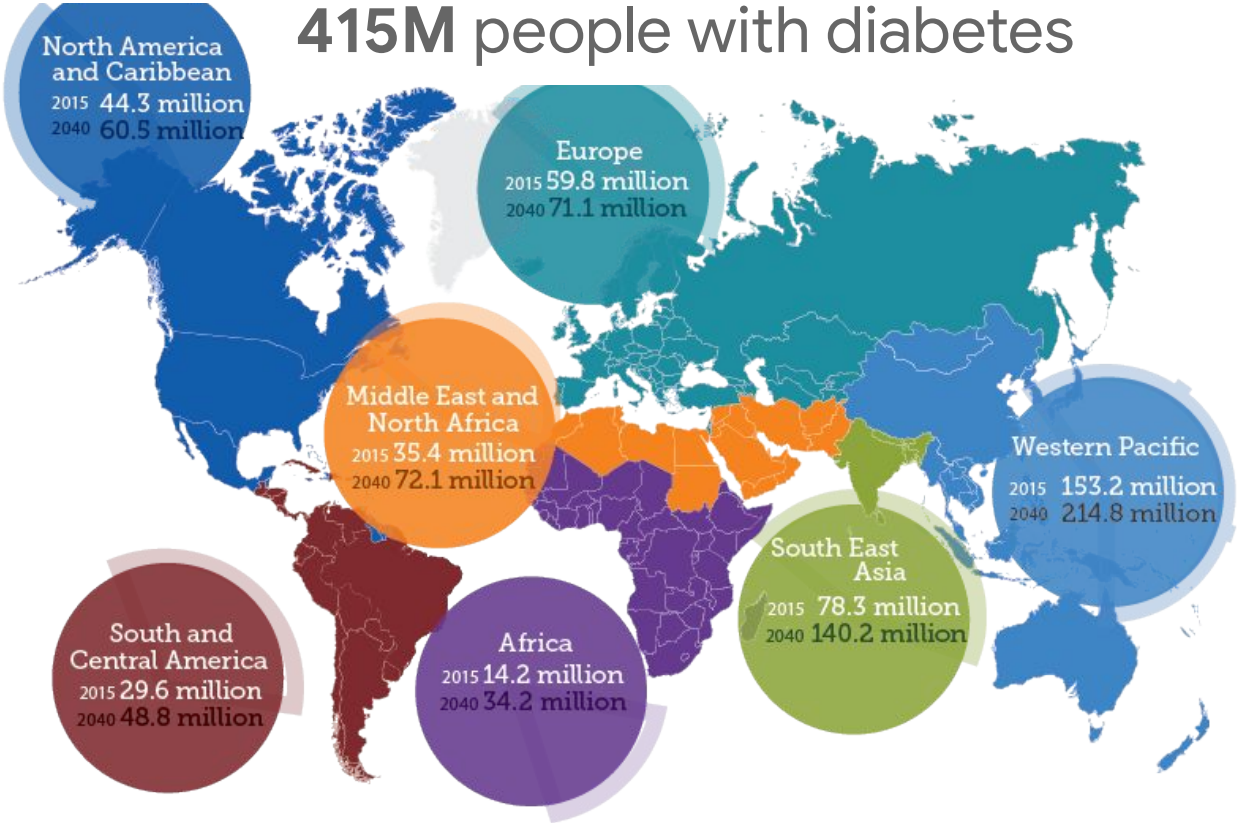
```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```



```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

```
model = keras.Sequential([keras.layers.Dense(units=1, input_shape=[1])])
model.compile(optimizer='sgd', loss='mean_squared_error')

xs = np.array([-1.0, 0.0, 1.0, 2.0, 3.0, 4.0], dtype=float)
ys = np.array([-3.0, -1.0, 1.0, 3.0, 5.0, 7.0], dtype=float)

model.fit(xs, ys, epochs=500)

print(model.predict([10.0]))
```

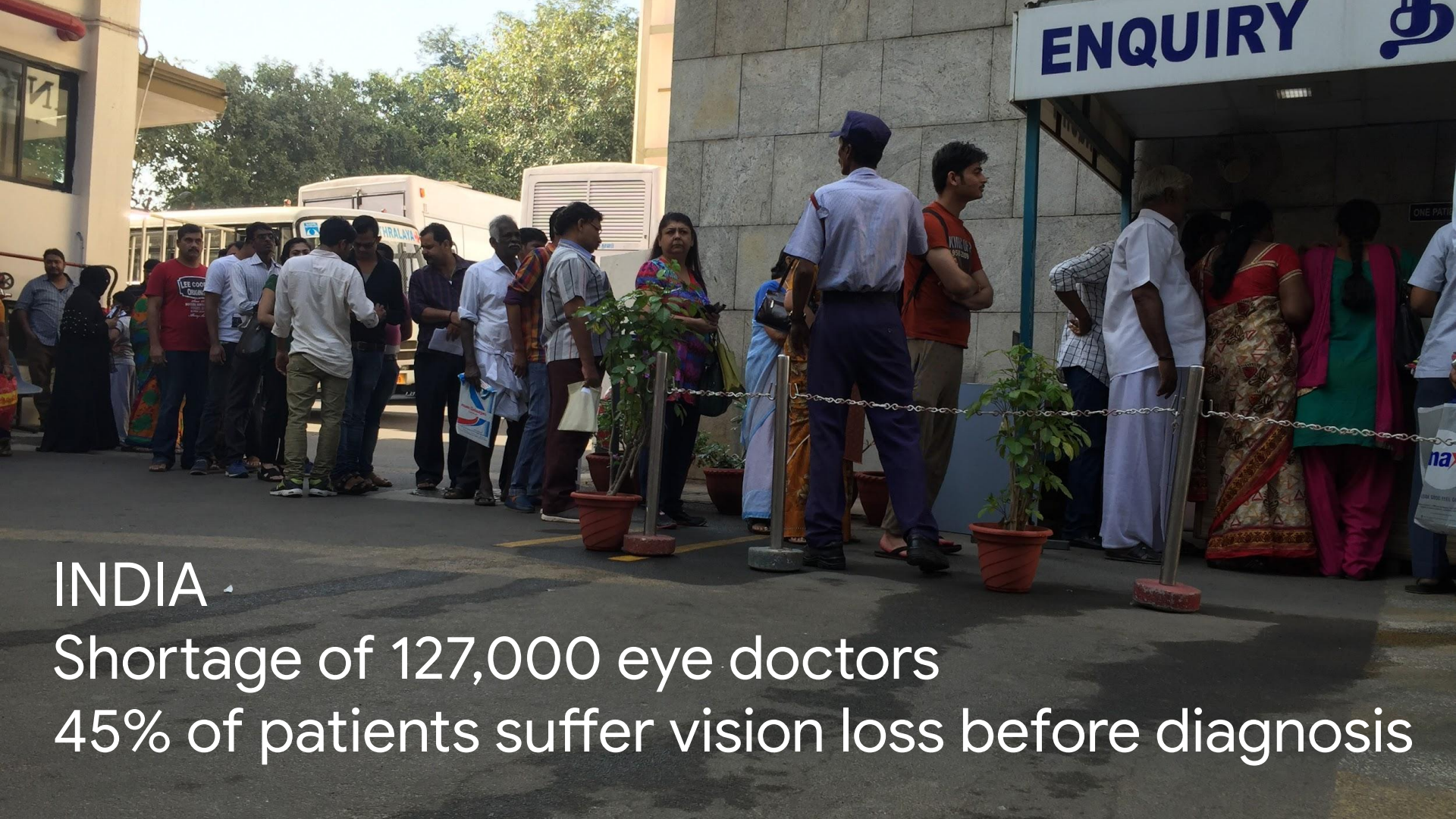
Diabetic retinopathy: fastest growing cause of blindness



Source: International Diabetes Federation 2015 Atlas, www.idf.org/e-library/epidemiology-research/diabetes-atlas/

Regular screening is key to preventing blindness

=



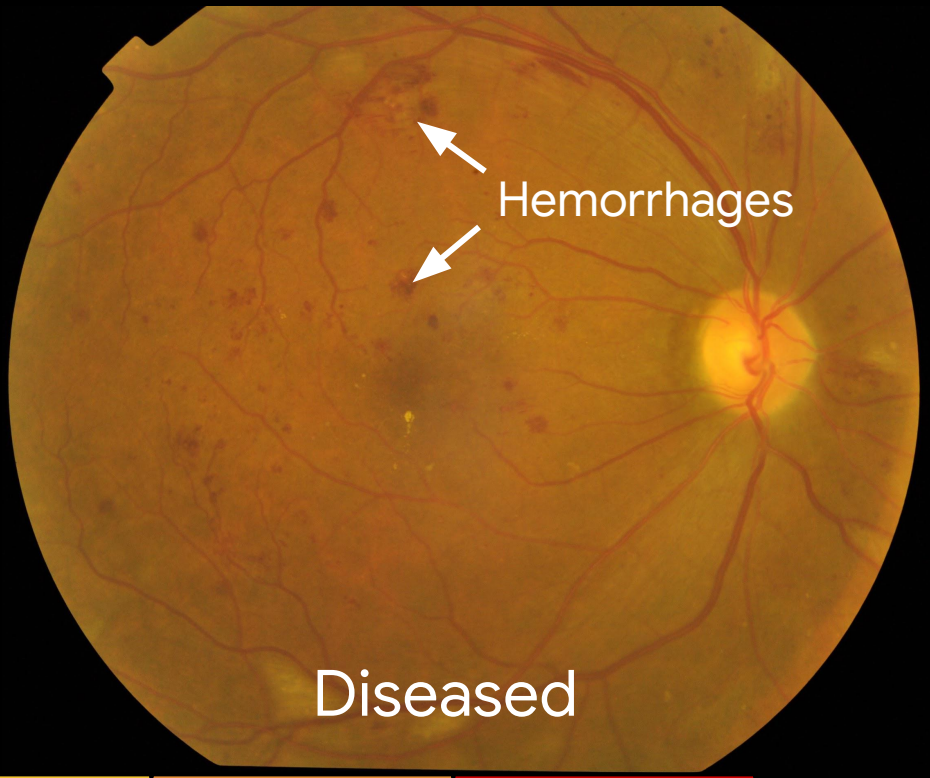
INDIA

Shortage of 127,000 eye doctors

45% of patients suffer vision loss before diagnosis

How DR is Diagnosed: Retinal Fundus Images

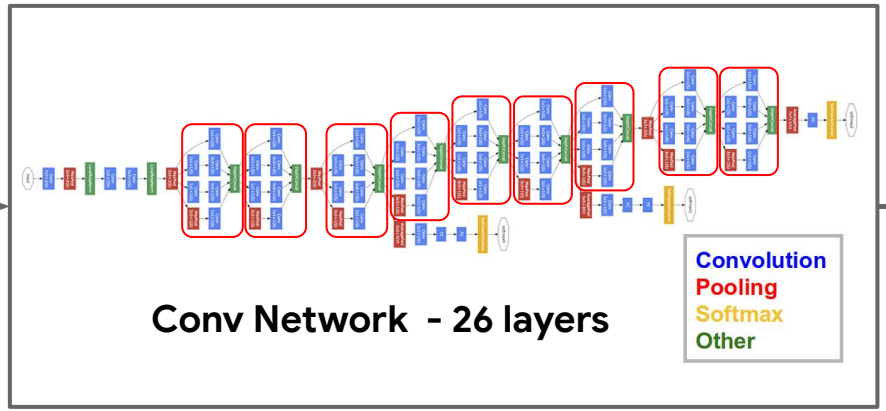
Healthy



Diseased

No DR	Mild DR	Moderate DR	Severe DR	Proliferative DR

Adapt deep neural network to read fundus images

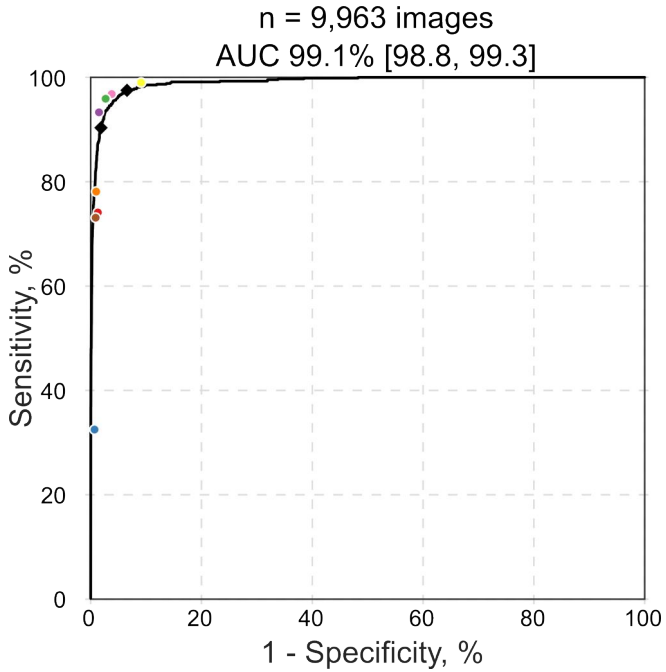


- No DR
- Mild DR
- Moderate DR
- Severe DR
- Proliferative DR
- Image Quality
- L/R eye
- Field of View

JAMA The Journal of the American Medical Association

JAMA | Original Investigation | INNOVATIONS IN HEALTH CARE DELIVERY

Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs



F-score

0.95	0.91
Algorithm	Ophthalmologist (median)

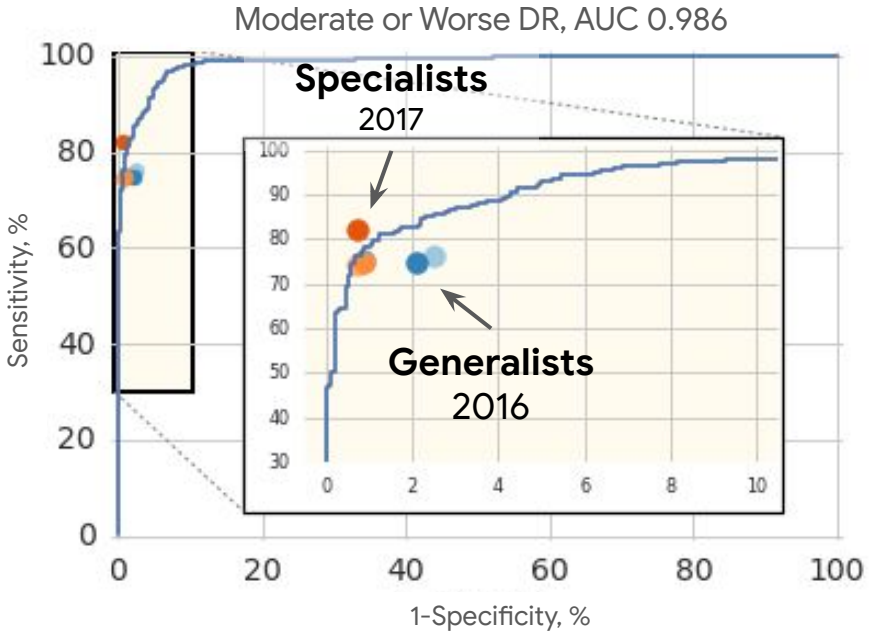
"The study by Gulshan and colleagues **truly represents the brave new world in medicine.**"

Dr. Andrew Beam, Dr. Isaac Kohane
Harvard Medical School

"Google just published this paper in JAMA (impact factor 37) [...] **It actually lives up to the hype.**"

Dr. Luke Oakden-Rayner
University of Adelaide

2016 - On Par with General Ophthalmologists
2017 - On Par with Retinal Specialist Ophthalmologists



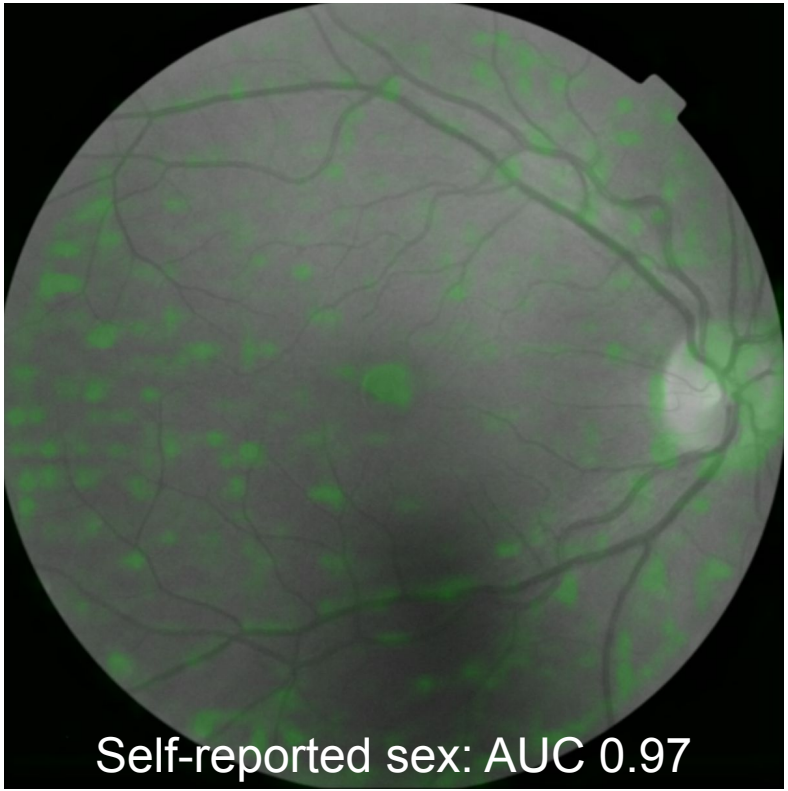
	Weighted Kappa
Ophthalmologists Individual	0.80-0.84
Algorithm	0.84
Retinal Specialists Individual	0.82-0.91

Grader variability and the importance of reference standards for evaluating machine learning models for diabetic retinopathy. J. Krause, *et al.*, *Ophthalmology*, doi.org/10.1016/j.ophtha.2018.01.034

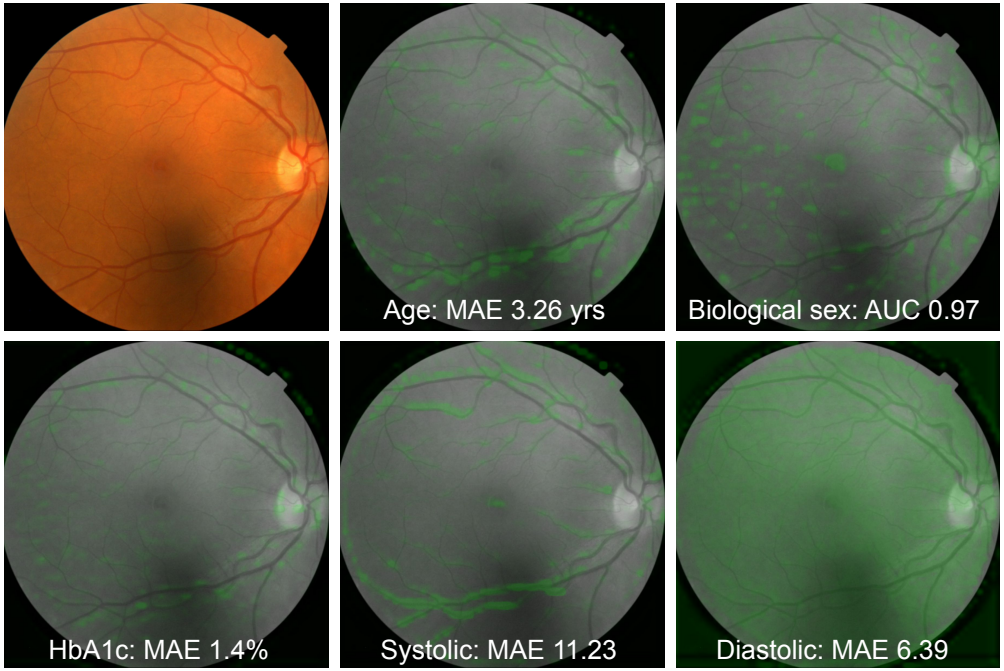
Completely new, novel scientific discoveries

Ophthalmologists can’t do this, so should be no better than flipping a coin (i.e. AUC of 0.50)

Completely new, novel scientific discoveries



Completely new, novel scientific discoveries



Predicting things that doctors can't predict from imaging

—

Potential as a new biomarker

Preliminary 5-yr MACE AUC: 0.7

—

Can we predict cardiovascular risk? If so, this is a very nice non-invasive way of doing so

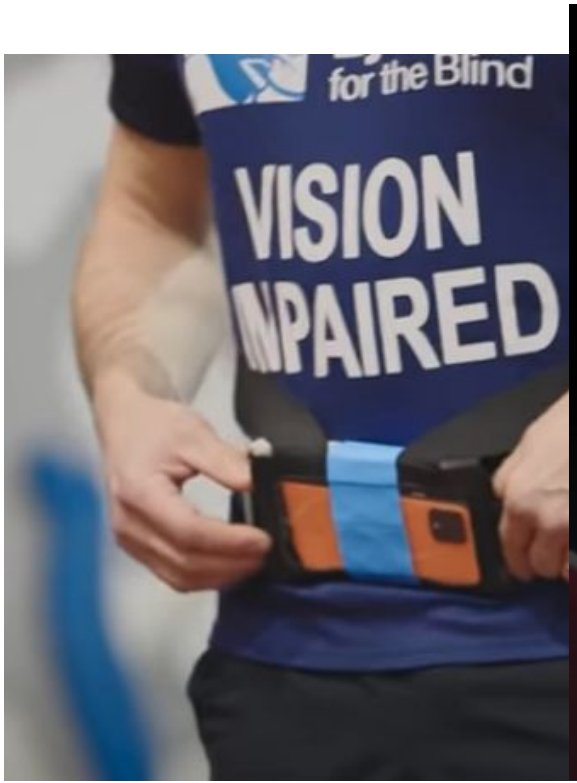
Can we also predict treatment response?

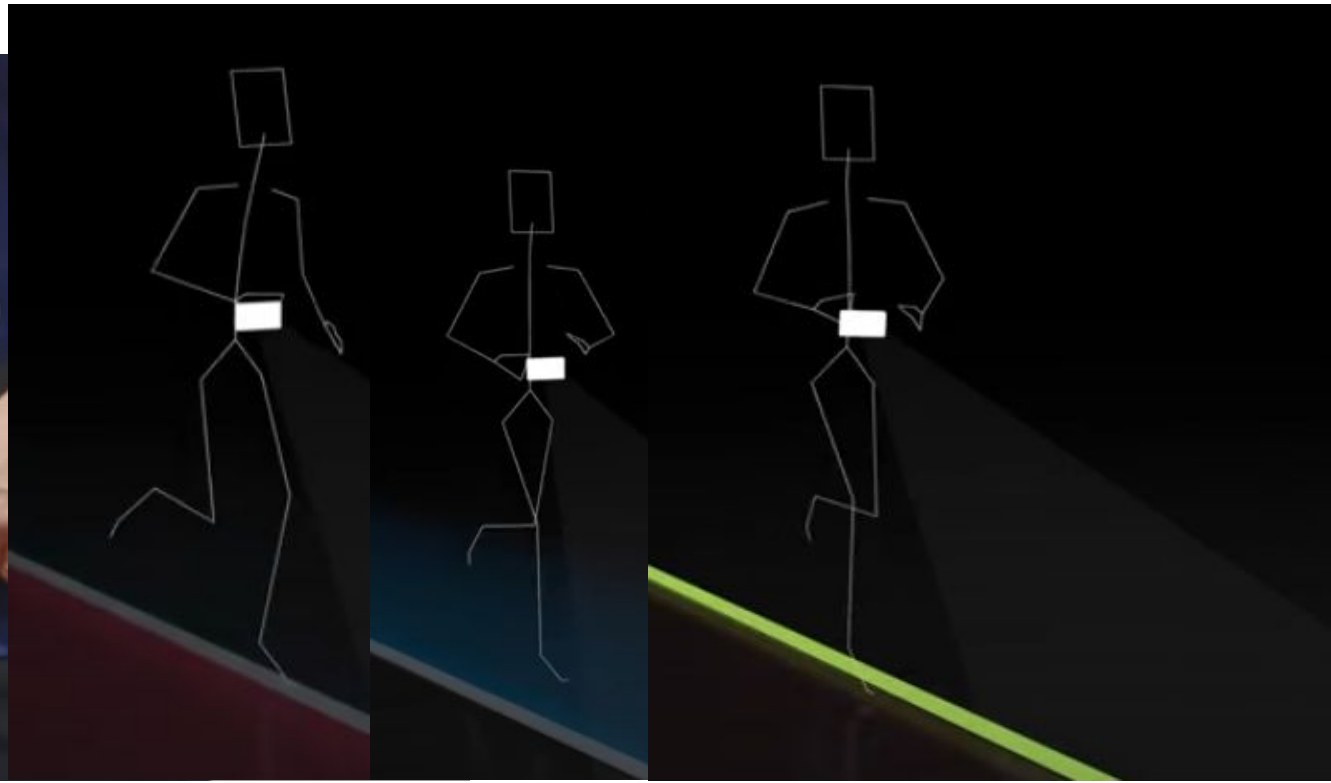
R. Poplin, A. Varadarajan *et al.* Predicting Cardiovascular Risk Factors from Retinal Fundus Photographs using Deep Learning. *Nature Biomedical Engineering, 2018.*

What’s next?

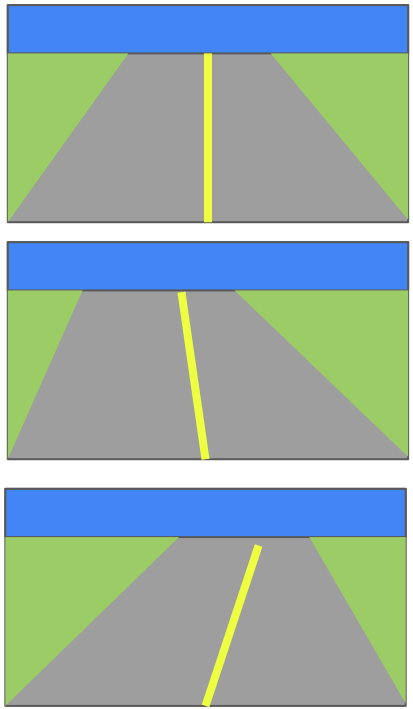
- AI Research continues to grow.
- Greater Cloud and AI collaboration
  - AI to be a significant driver in Cloud Solution adoption
- IT Problem Detection and Avoidance
- AI and ML Ops

Project Guideline

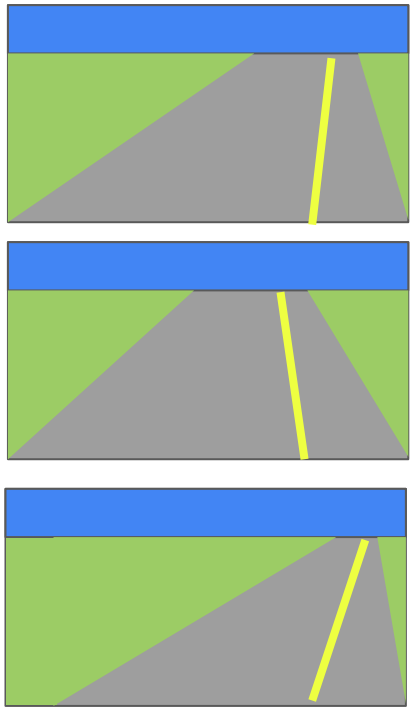




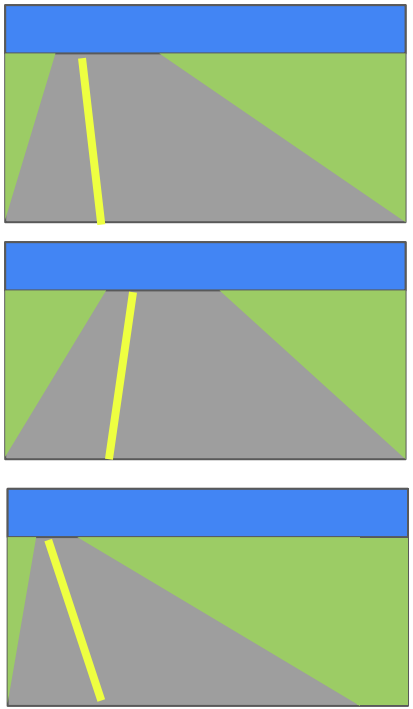
How would it work?



Good

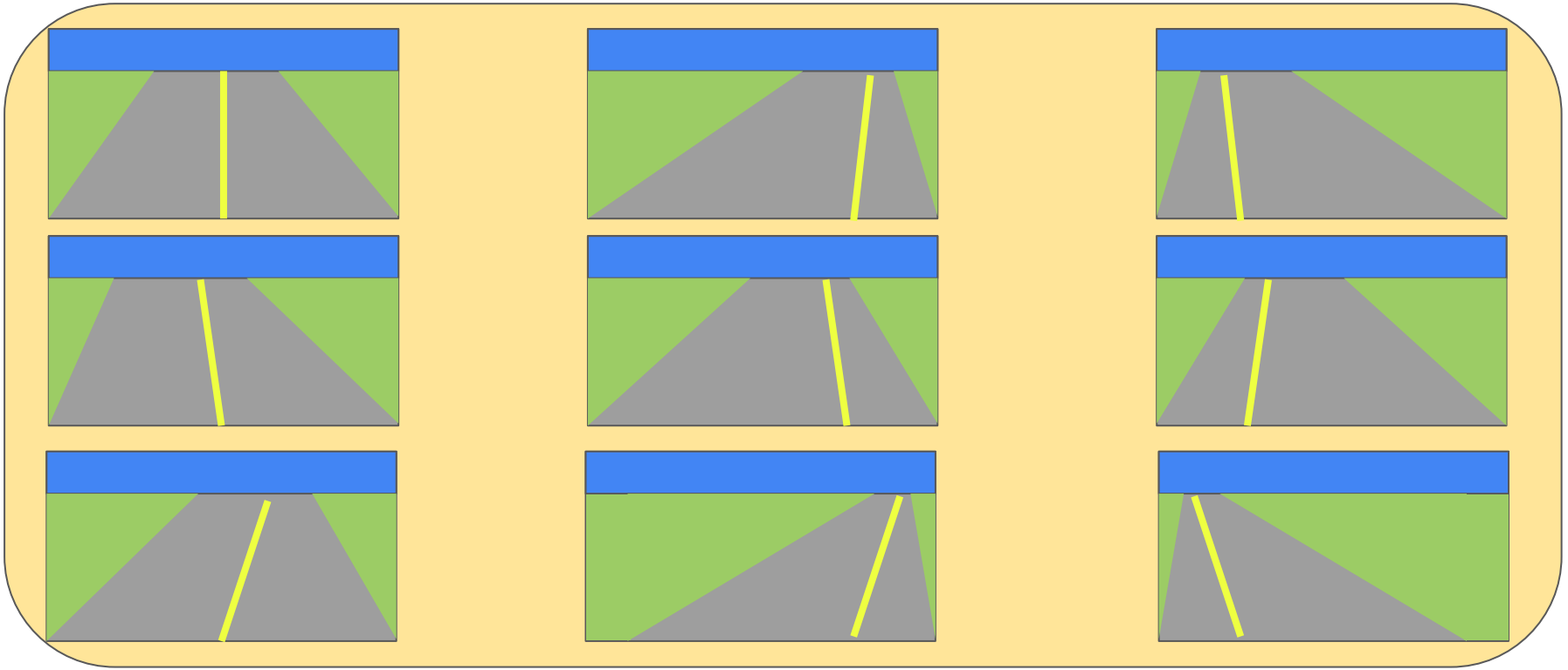


Move Right!



Move Left!

How would it work?



Good

Move Right!

Move Left!

How would it work?

Good

Move Right!

Move Left!