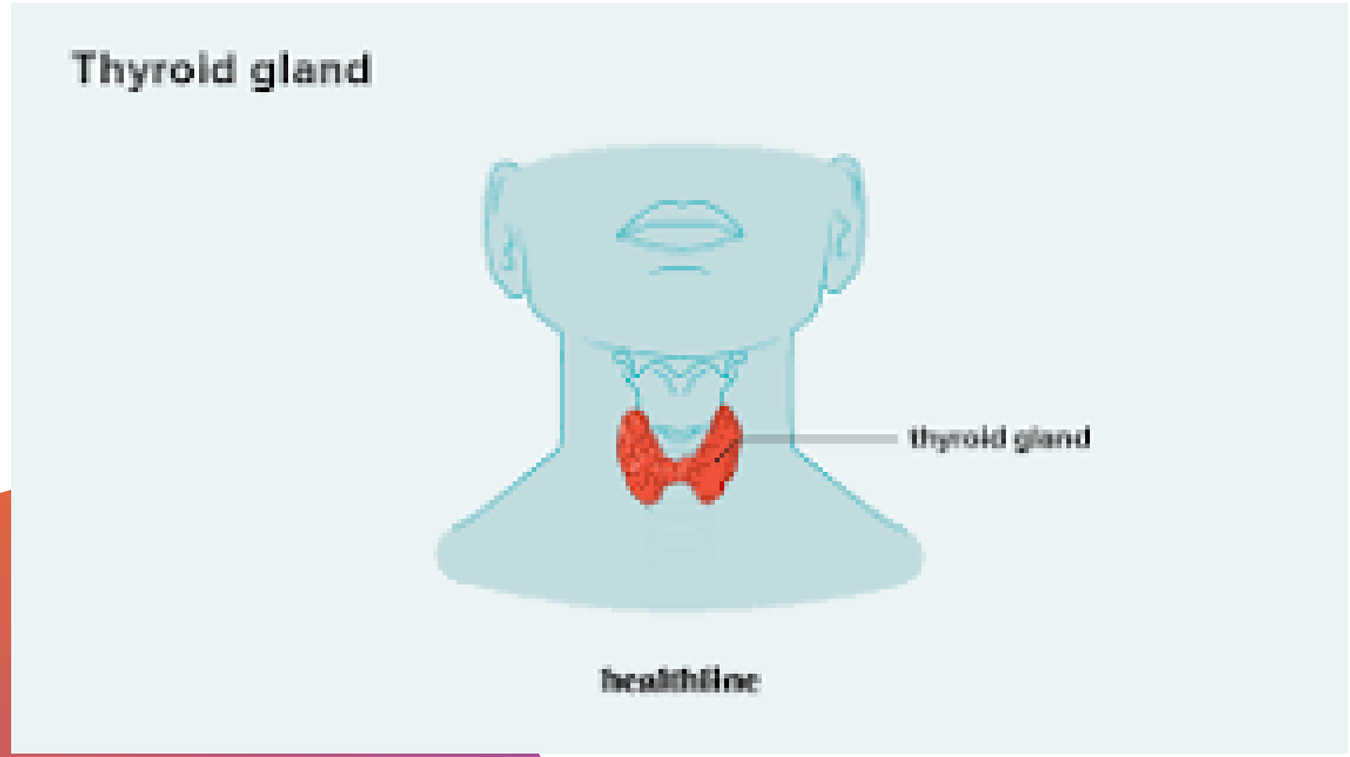
Summer Project

Thyroid Prediction and Risk Classification

What is the thyroid gland?



Butterfly-shaped gland near Adam's Apple

What are thyroid anomalies?

Hyperthyroidism
and Hypothyroidism

What is ACR TI-RADS?
What does it tell us?

Risk reporting system for thyroid ultrasound
Tells us what course of action to take

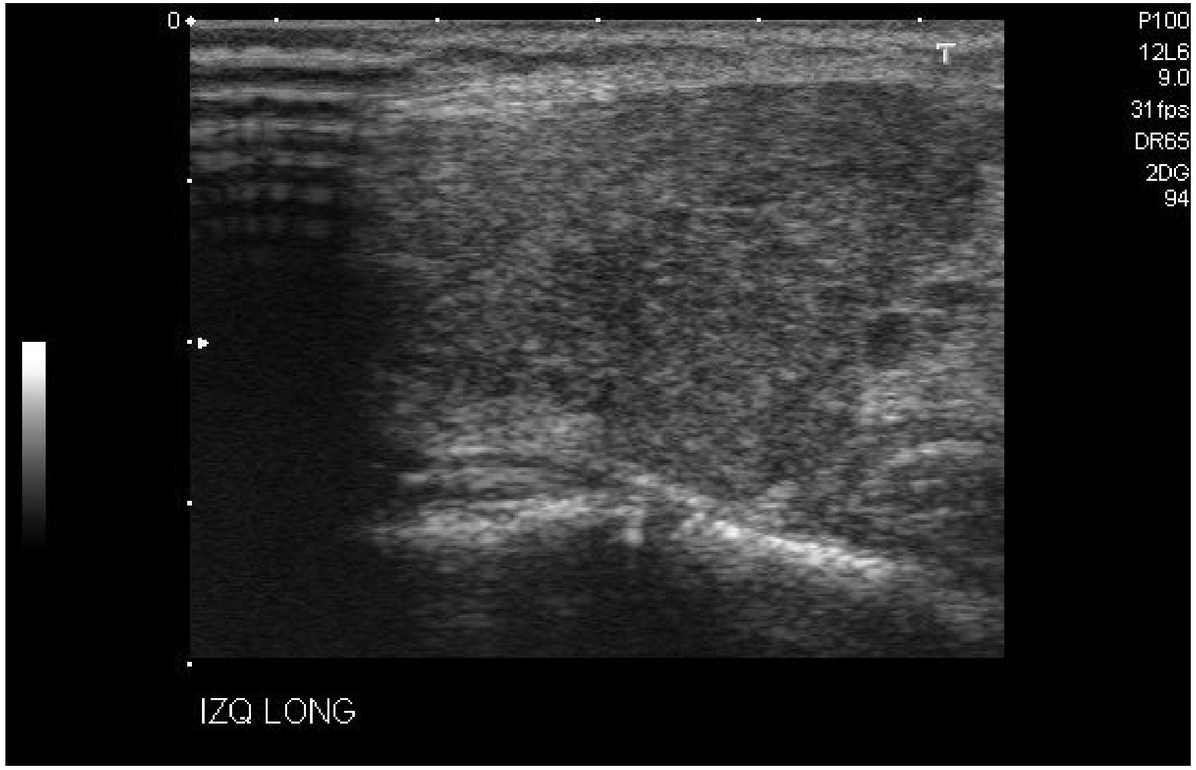
Why do we need DL for this situation?

Radiologists who interpret thyroid ultrasonography (US) images frequently face the dilemma of how to report nodules, which are extremely common and overwhelmingly benign (1). Like risk-stratification systems from other professional societies and investigators, the American College of Radiology (ACR) Thyroid Imaging Reporting and Data System (TI-RADS) aims to provide an easy-to-apply method for practitioners to determine management (2). We believe that this will improve consistency across practices and institutions and will benefit patients by applying guidelines that are based on evidence and consensus expert opinion.

In ACR TI-RADS, nodules classified as spongiform are not subject to further feature assignment and are treated as benign, with no further follow-up needed. Most investigators agree that spongiform refers to the presence of very small cysts that are akin to the fluid-filled spaces in a wet sponge, but there is some controversy as to how much of the nodule must have this appearance to qualify (6,7). Per the ACR thyroid lexicon, at least 50% of the nodule's volume should be occupied by tiny cysts (Fig 3) (5). It should be

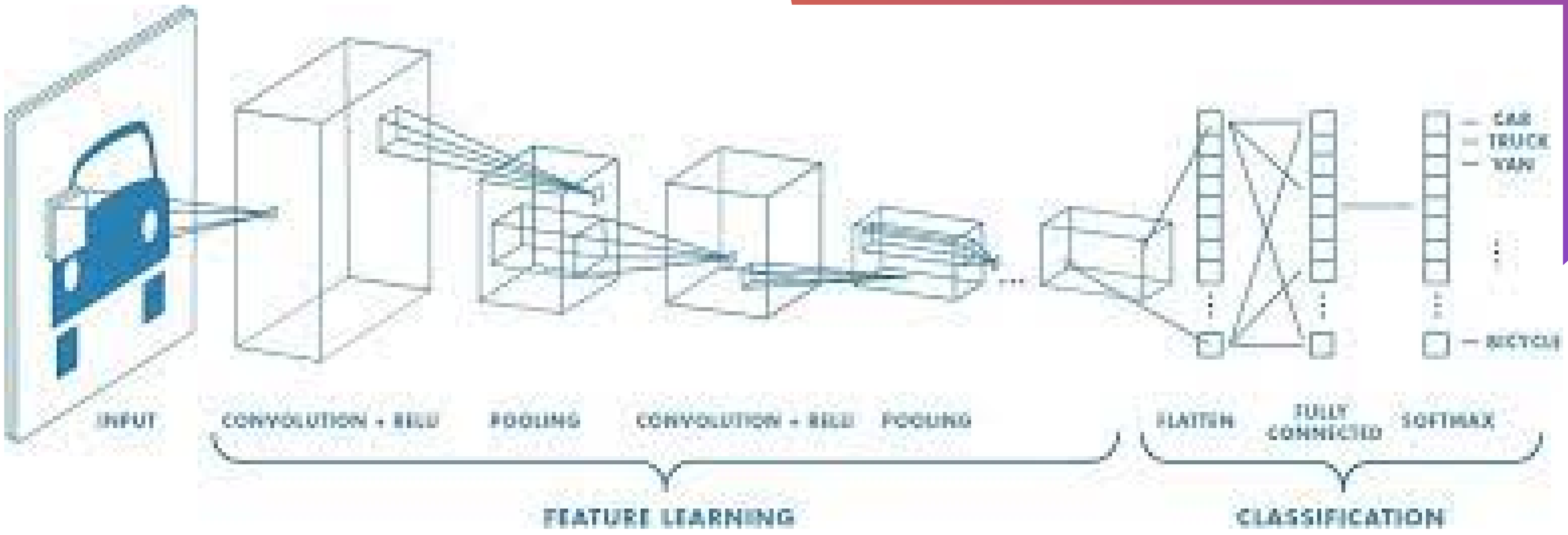
Distinguishing solid nodules from mixed cystic and solid nodules may be difficult in practice, as they represent a continuum. Unlike with spongiform nodules, ACR TI-RADS does not require that the observer estimate the percentage of a nodule that is solid, as this determination is often highly subjective and is less important than the characteristics of the solid component. This represents a departure from the lexicon, which explicitly describes predominately cystic and predominately solid nodules (5). As a general principle, however, otherwise-solid nodules that contain small cystic components that occupy no more than approximately 5% of the overall volume should be classified as solid (Fig 4).

more worrisome (10,11). Position and shape do not contribute to the nodule's point total, but if the solid component exhibits any of these suspicious features, we occasionally recommend FNA even if the nodule does not otherwise meet criteria for biopsy. The presence of flow at color or power Doppler imaging does not reliably indicate that the solid component is malignant, nor does its absence mean that it is benign. However, when seen, vascularity shows that the solid material represents viable tissue rather than blood clot, debris, or necrotic tissue.

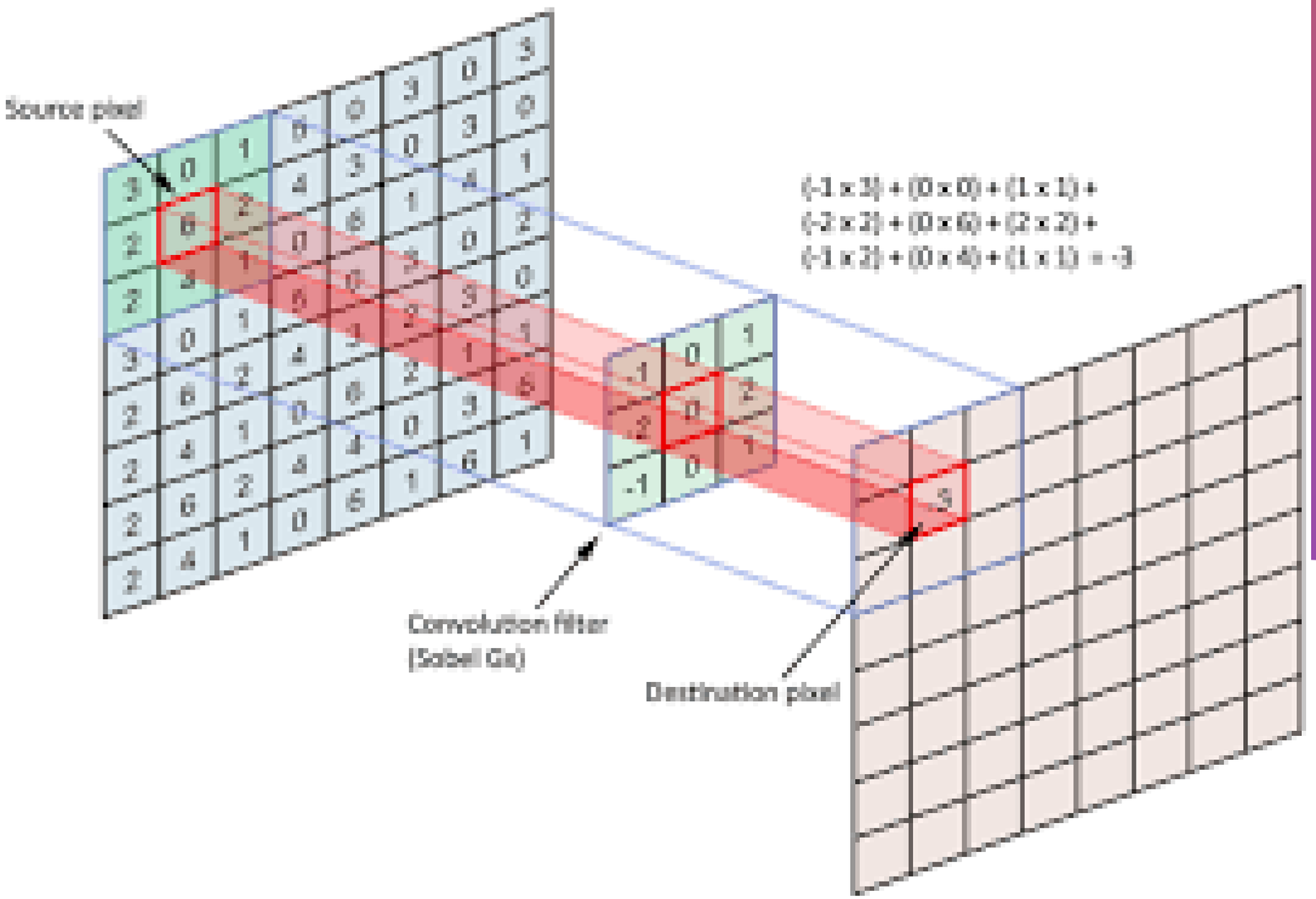

0
T
P100
12L6
9.0
31fps
DR65
2DG
94
IZQ LONG

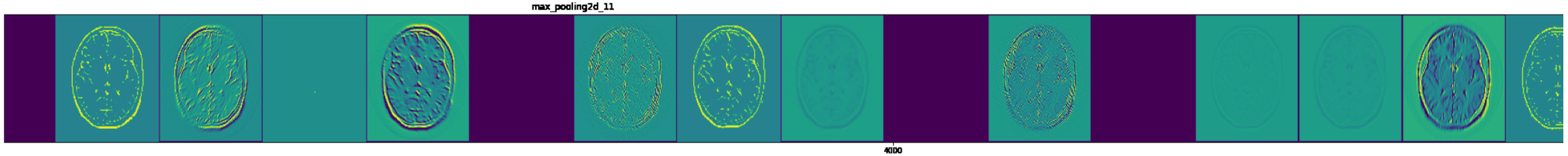

T
P100
12L6
12.0
31fps
DR65
2DG
92
LOBULO IZQUIERDO
TIRADS 5 (4C)

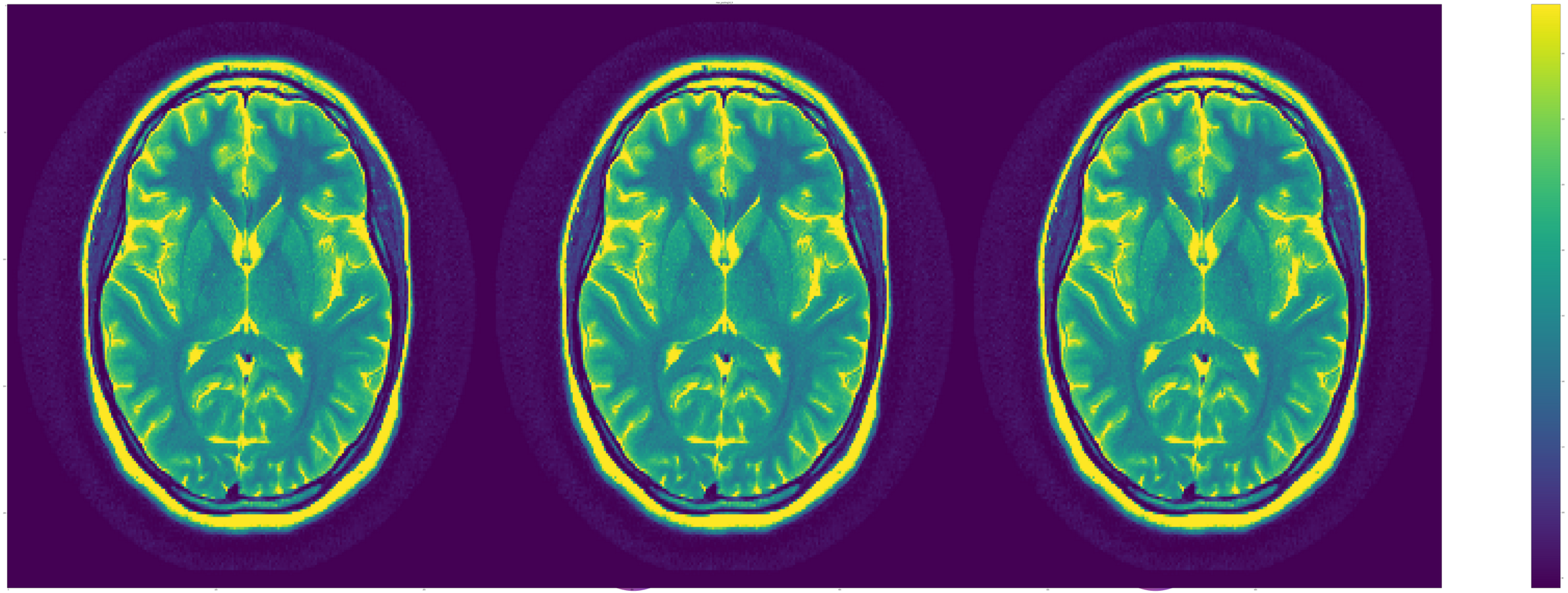
What is a CNN?

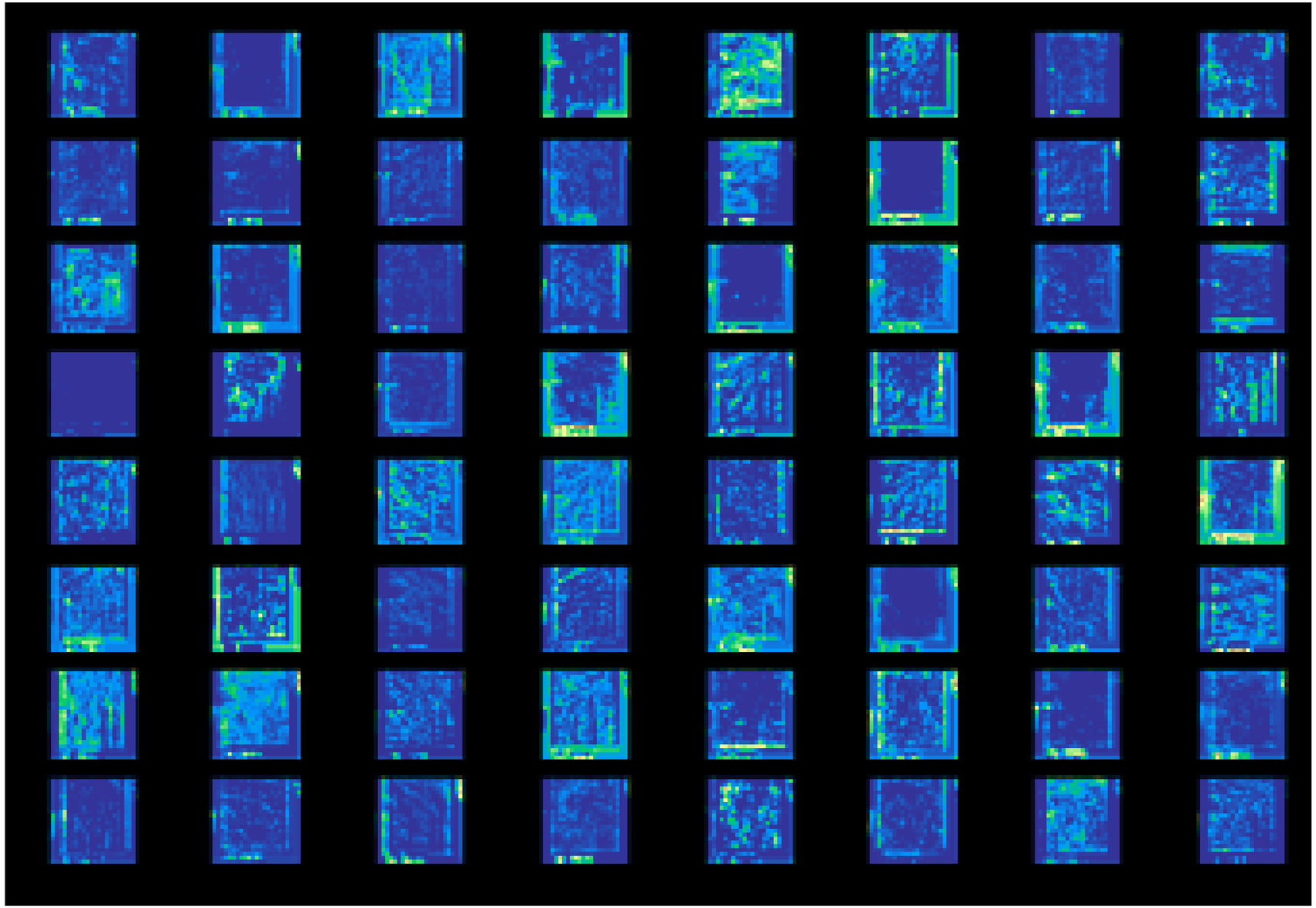

CAR
TRUCK
VAN
BICYCLE
INPUT
CONVOLUTION + RELU
POOLING
CONVOLUTION + RELU
POOLING
FLATTEN
FULLY CONNECTED
SOFTMAX
FEATURE LEARNING
CLASSIFICATION

What is a convolution?

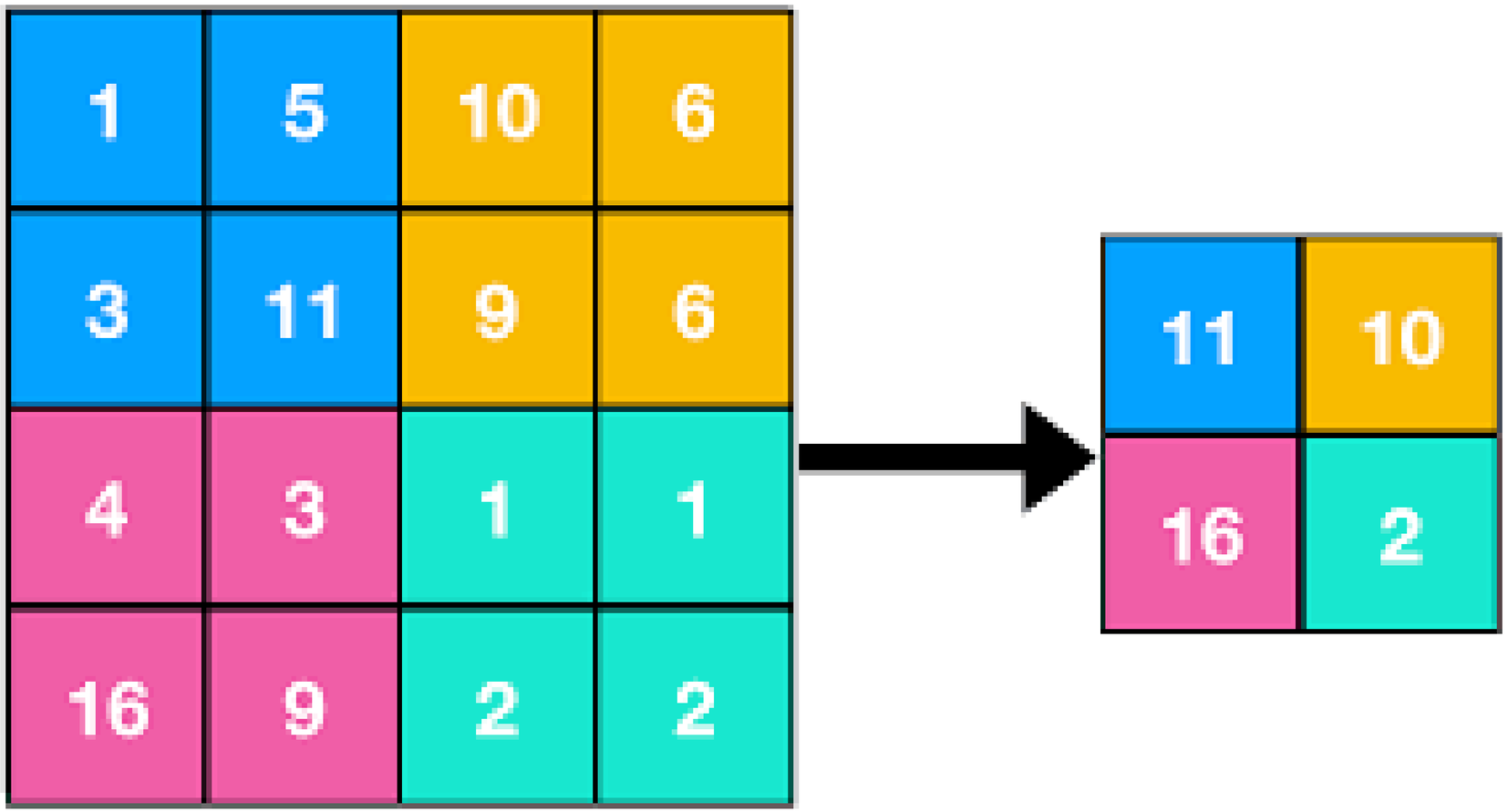

Source pixel
(-1 x 3) + (0 x 0) + (1 x 1) +
(-2 x 2) + (0 x 6) + (2 x 2) +
(-1 x 2) + (0 x 4) + (1 x 1) = -3
Convolution filter
(Sobel Gx)
Destination pixel


max_pooling2d_11
4000





What is pooling?


1
5
10
6
3
11
9
6
4
3
1
1
16
9
2
2
11
10
16
2

What is transfer learning?

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task.

Why is VGG16 better than others?

The VGG16 model achieves almost 92.7% top-5 test accuracy in ImageNet. ImageNet is a dataset consisting of more than 14 million images belonging to nearly 1000 classes.

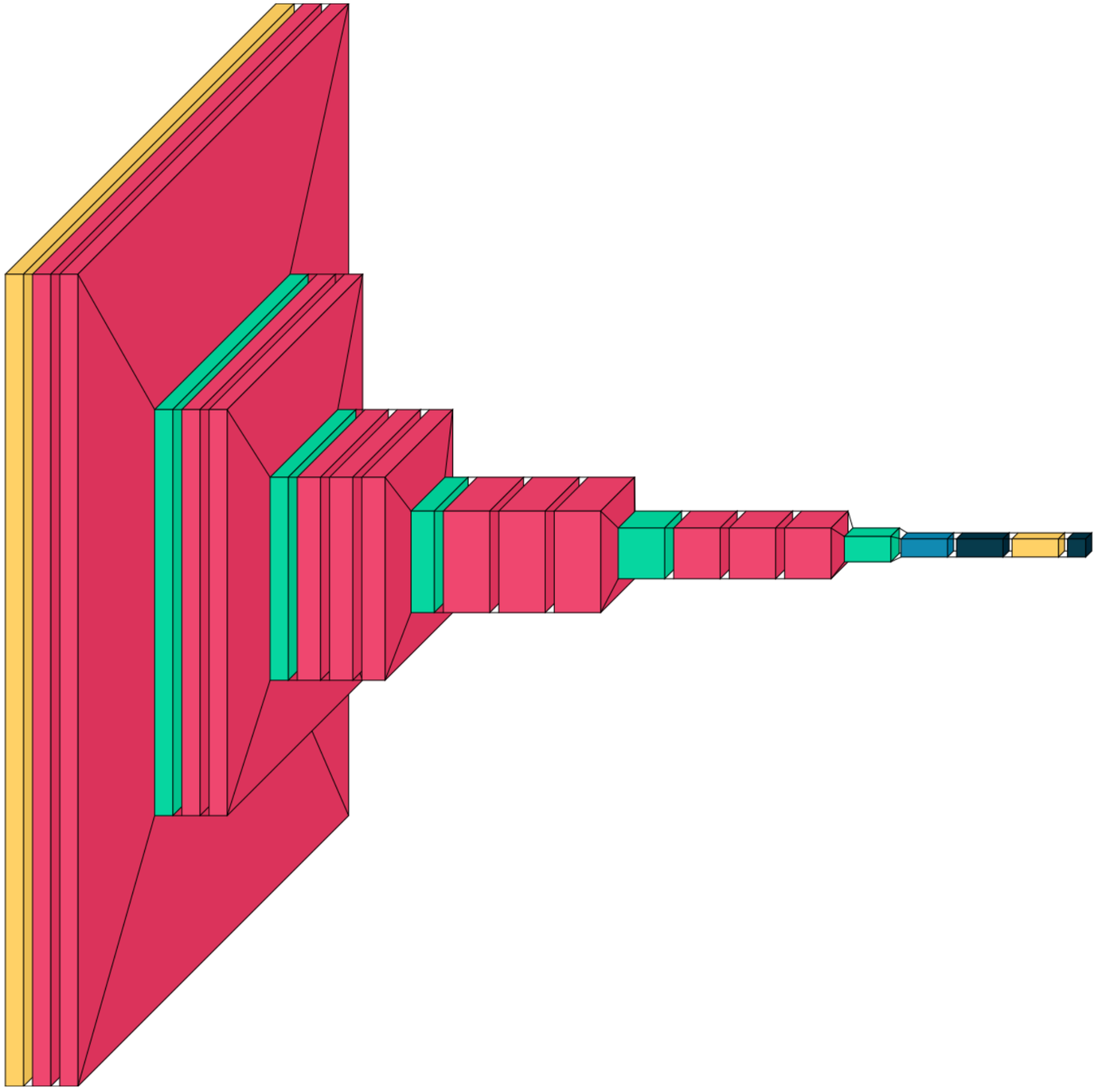
FIRST MODEL

Dataset Information

551 labeled images
273 training images
218 malignant 55 benign
139 labeled images
105 malignant 34 benign

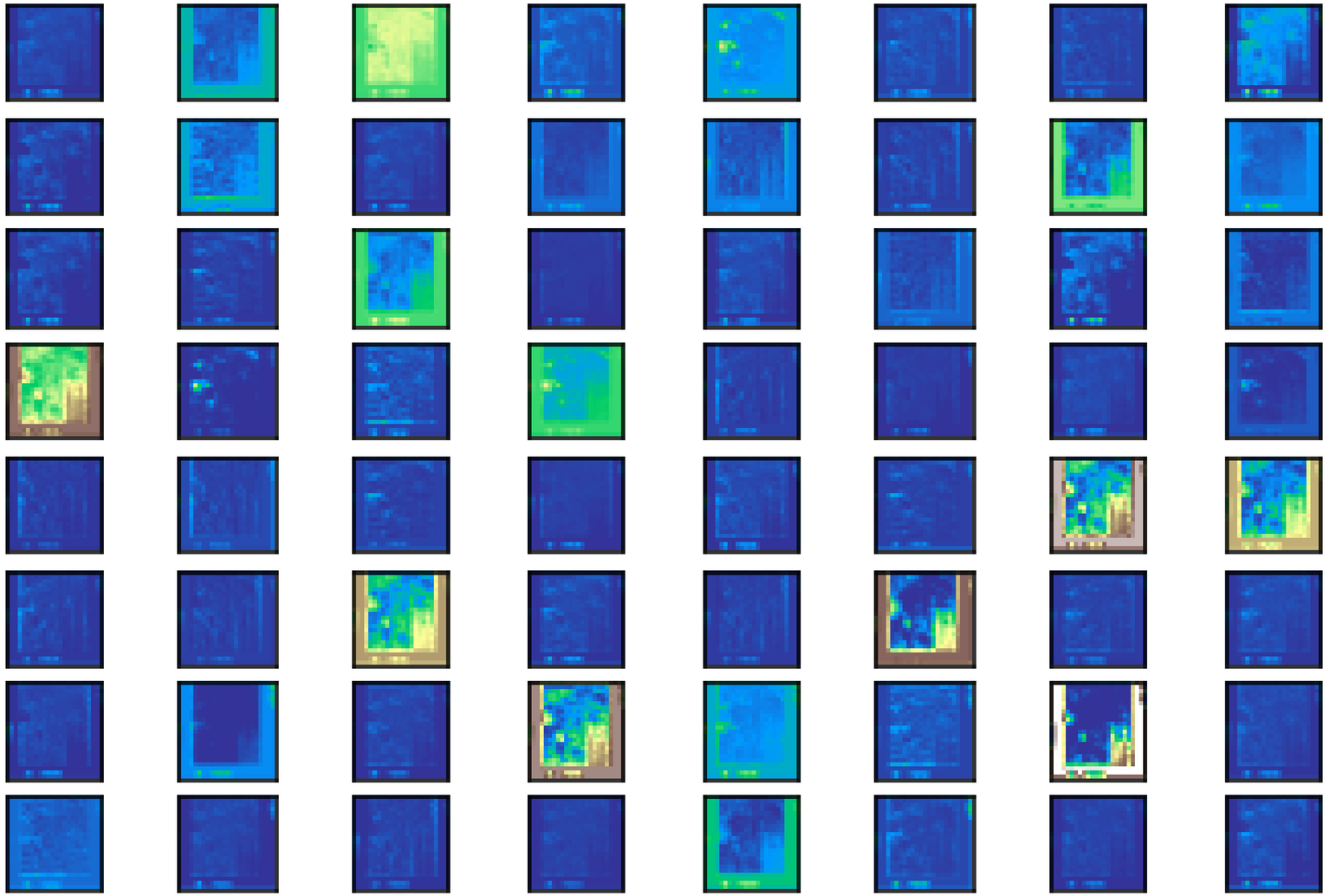
Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808

```
block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

flatten (Flatten)           (None, 25088)             0

fc1 (Dense)                 (None, 4096)              102764544

fc2 (Dense)                 (None, 4096)              16781312

predictions (Dense)         (None, 1000)              4097000
```



```
es = EarlyStopping(monitor='acc', mode='min', verbose=1, patience=50,baseline=0.7)

mc = ModelCheckpoint('bestvgg_model.h5', monitor='acc', mode='max', save_best_only=True)
```

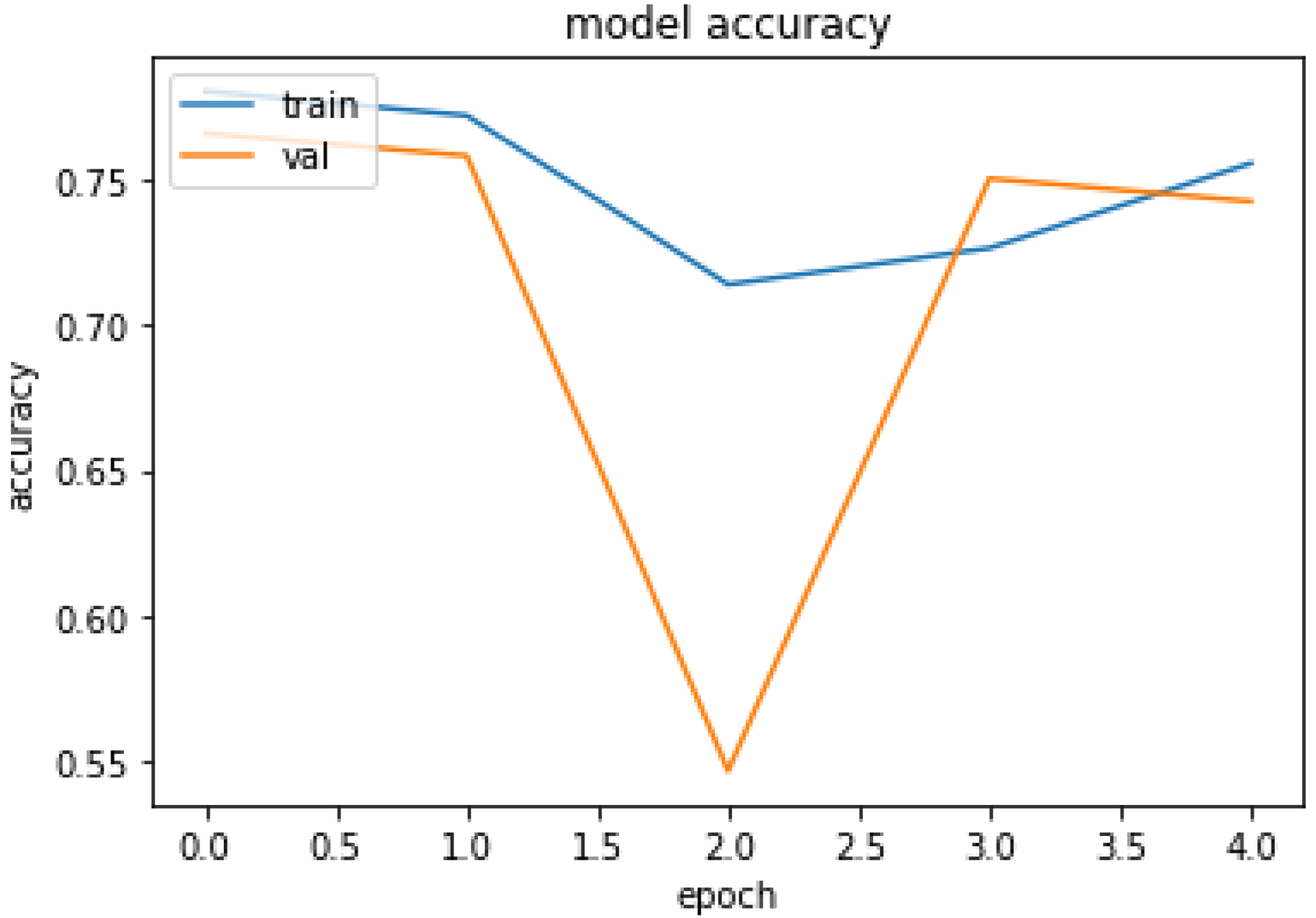
```
Epoch 1/5
8/8 [==============================] - 194s 25s/step - loss: 0.4599 - acc: 0.7801 - val_loss: 0.6975 - val_acc: 0.7656
Epoch 2/5
8/8 [==============================] - 186s 24s/step - loss: 0.7531 - acc: 0.7718 - val_loss: 0.7441 - val_acc: 0.7578
Epoch 3/5
8/8 [==============================] - 184s 24s/step - loss: 0.6988 - acc: 0.7137 - val_loss: 0.6663 - val_acc: 0.5469
Epoch 4/5
8/8 [==============================] - 185s 25s/step - loss: 0.6318 - acc: 0.7261 - val_loss: 0.5924 - val_acc: 0.7500
Epoch 5/5
8/8 [==============================] - 184s 24s/step - loss: 0.5752 - acc: 0.7552 - val_loss: 0.5669 - val_acc: 0.7422
```



What is accuracy?

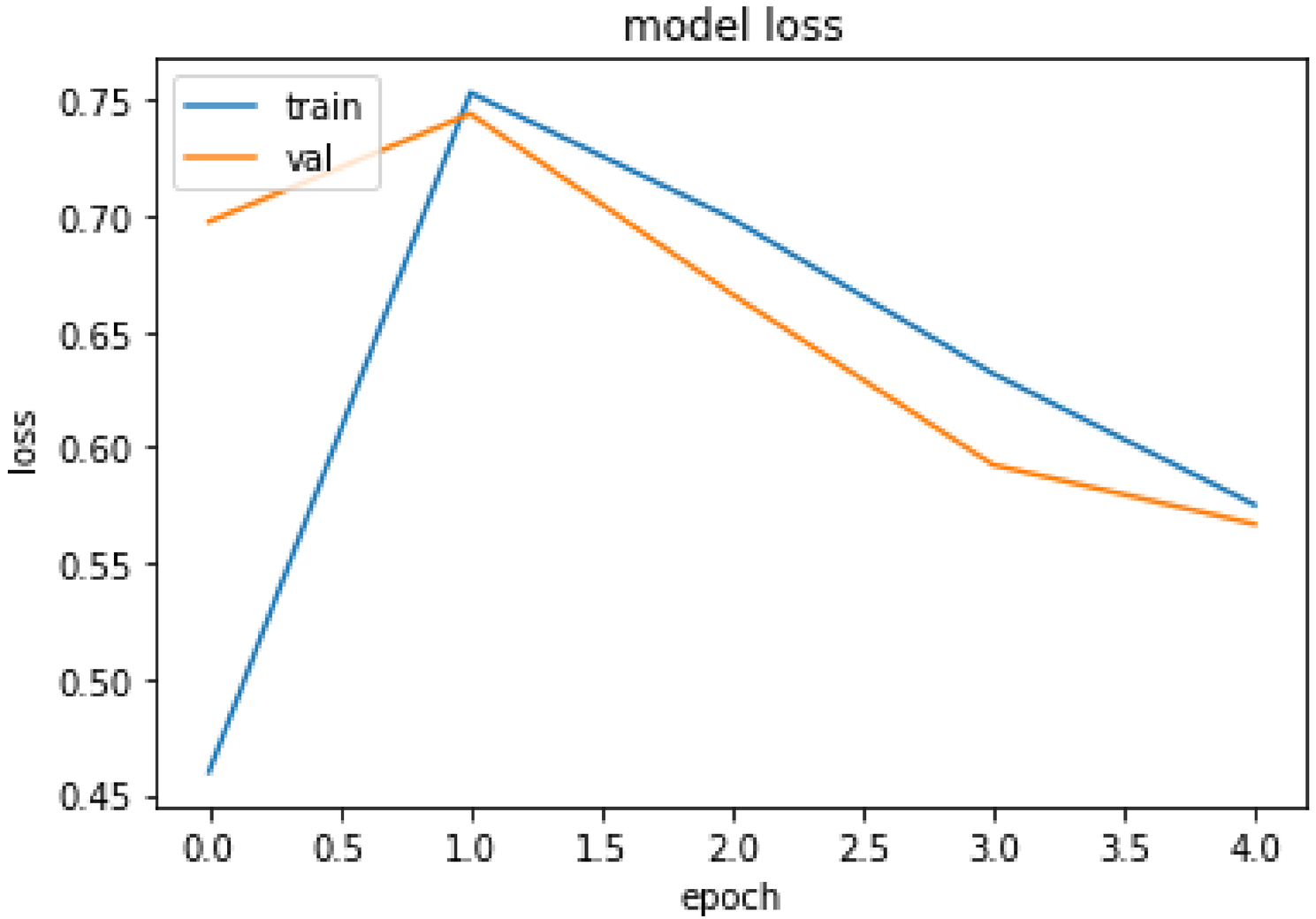
Accuracy is one metric for evaluating classification models.

Informally, accuracy is the fraction of predictions our model got right.


model accuracy
train
val
accuracy
0.75
0.70
0.65
0.60
0.55
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0
epoch

What is loss?

Loss is a measurement of how good your model is in terms of predicting the expected outcome.

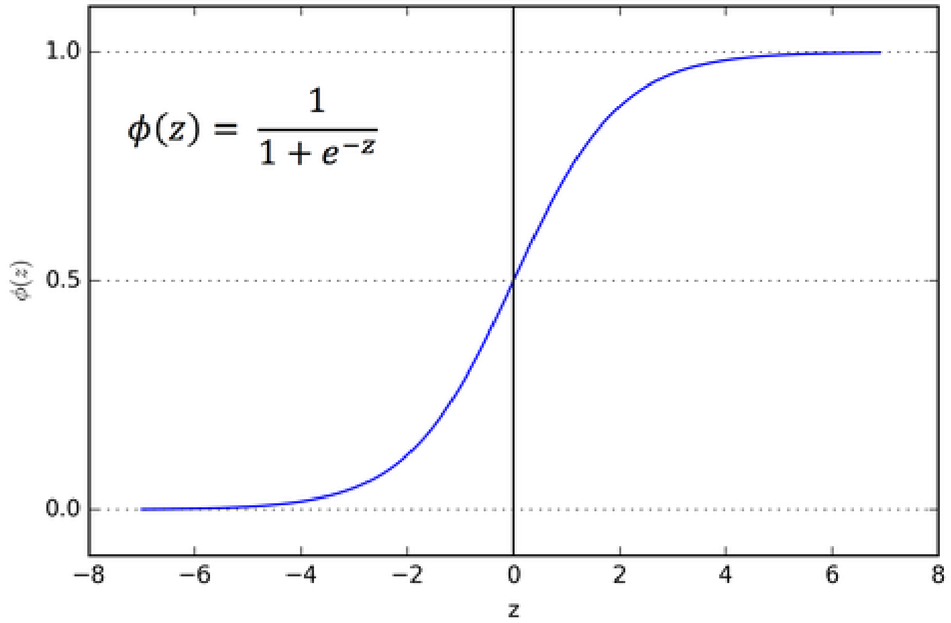

model loss
train
val
loss
epoch
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

How to make use of VGG16 for our custom dataset?

```
dense (Dense)                (None, 512)               262656

dropout (Dropout)            (None, 512)               0

dense_1 (Dense)              (None, 2)                 1026
```

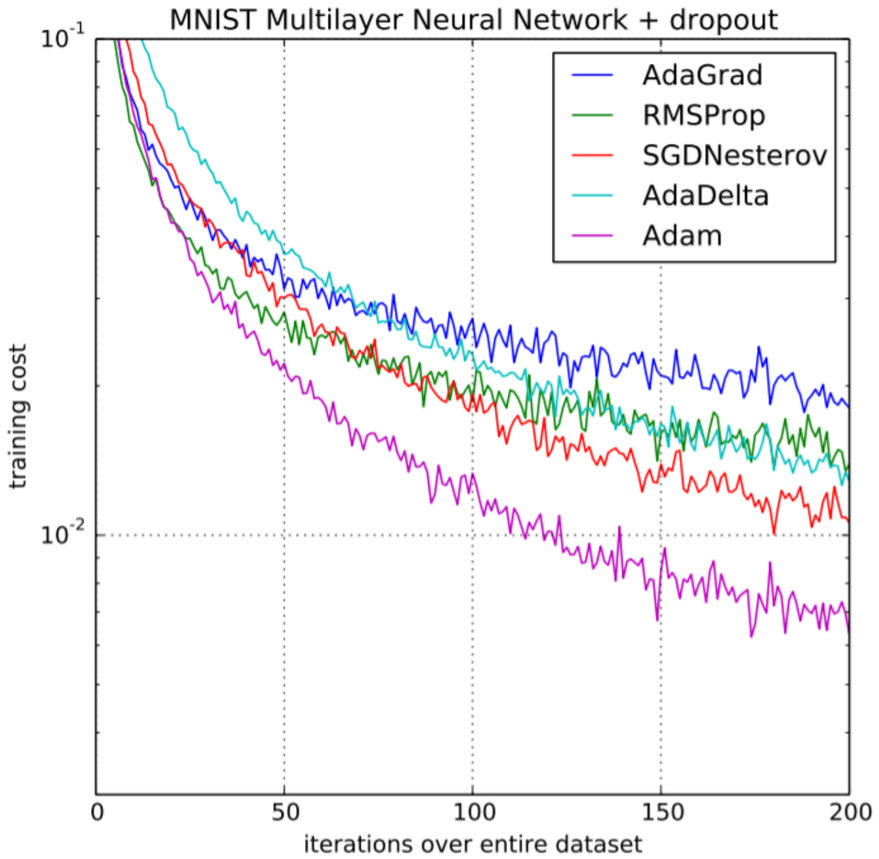
Why should we use VGG16? Why can't we train our own CNNs?

Why did we use sigmoid activation function here?



What does image augmentation do here?

Why did we use Adam optimiser?



SECOND MODEL

Dataset Information

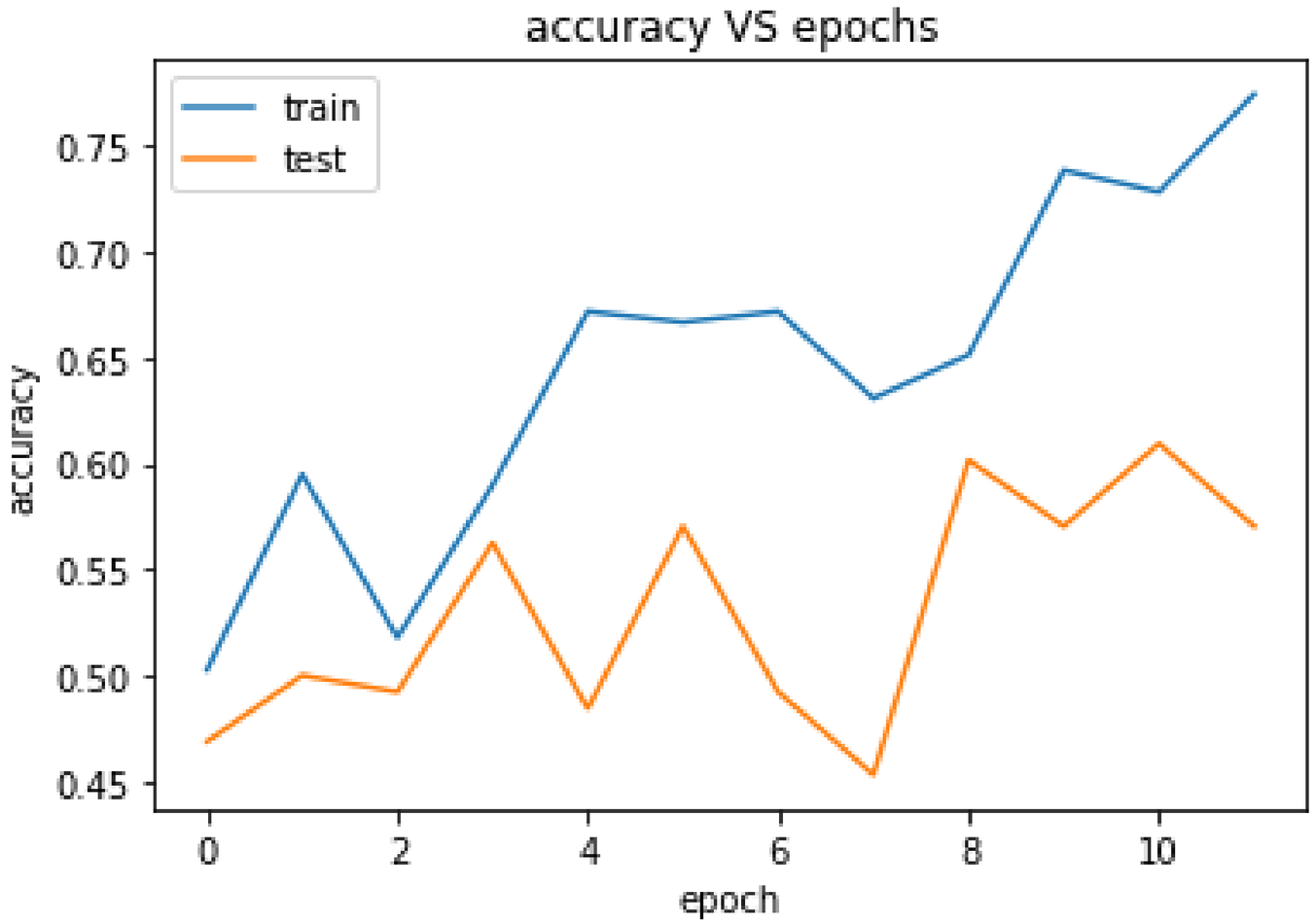
195 images belonging to 5 classes
72, 39, 19, 34, 31 images

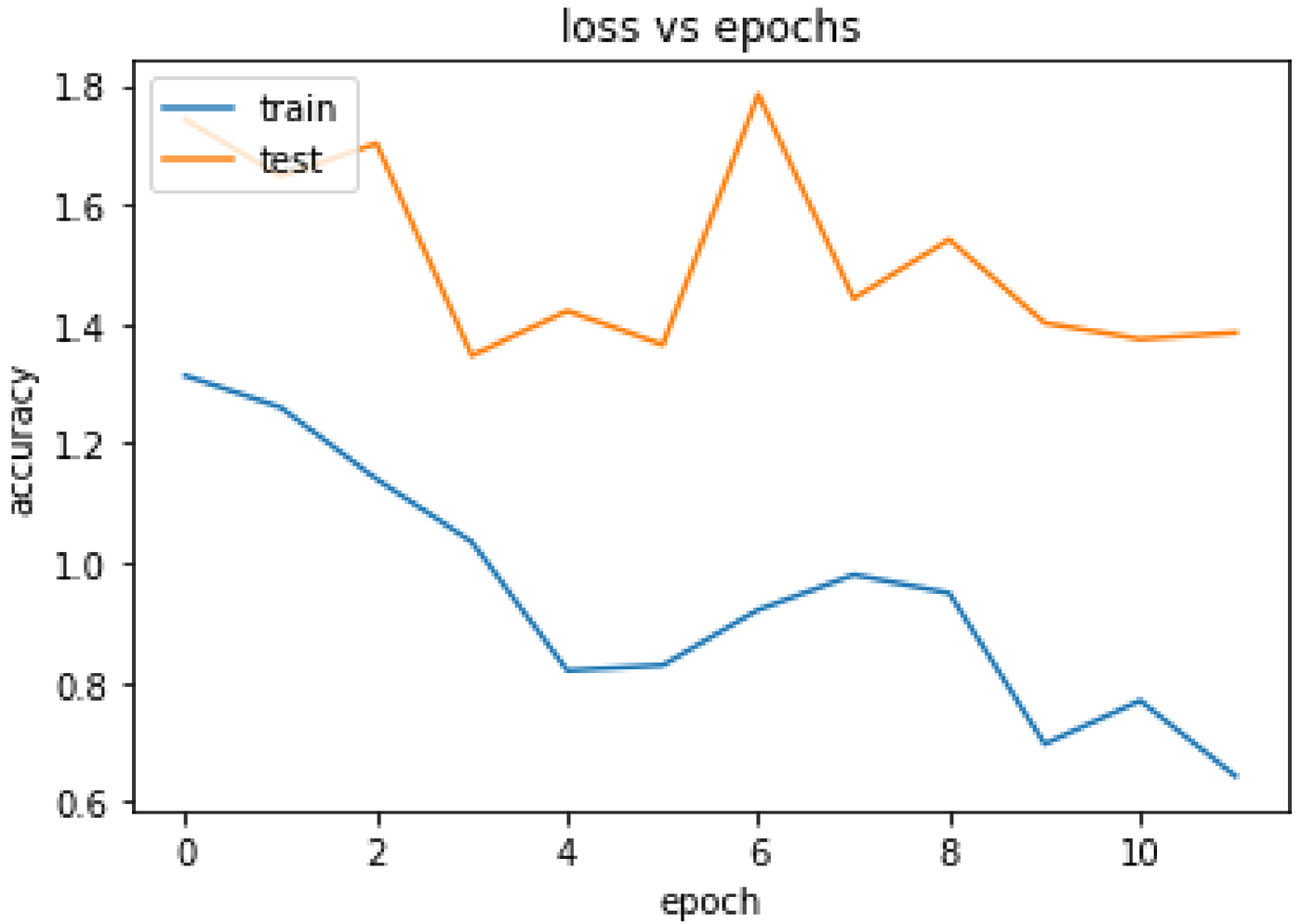
128 images belonging to 5 classes
34, 23, 10, 36 and 25 files

Layer (type)	Output Shape	Param #
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
block1_conv1 (Conv2D)	(None, 224, 224, 64)	1792
block1_conv2 (Conv2D)	(None, 224, 224, 64)	36928
block1_pool (MaxPooling2D)	(None, 112, 112, 64)	0
block2_conv1 (Conv2D)	(None, 112, 112, 128)	73856
block2_conv2 (Conv2D)	(None, 112, 112, 128)	147584
block2_pool (MaxPooling2D)	(None, 56, 56, 128)	0
block3_conv1 (Conv2D)	(None, 56, 56, 256)	295168
block3_conv2 (Conv2D)	(None, 56, 56, 256)	590080
block3_conv3 (Conv2D)	(None, 56, 56, 256)	590080
block3_pool (MaxPooling2D)	(None, 28, 28, 256)	0
block4_conv1 (Conv2D)	(None, 28, 28, 512)	1180160
block4_conv2 (Conv2D)	(None, 28, 28, 512)	2359808

```
block4_conv3 (Conv2D)       (None, 28, 28, 512)       2359808

block4_pool (MaxPooling2D)  (None, 14, 14, 512)       0

block5_conv1 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv2 (Conv2D)       (None, 14, 14, 512)       2359808

block5_conv3 (Conv2D)       (None, 14, 14, 512)       2359808

block5_pool (MaxPooling2D)  (None, 7, 7, 512)         0

flatten (Flatten)           (None, 25088)             0

dense (Dense)               (None, 5)                 125445
```

```
Epoch 1/12
7/7 [==============================] - 157s 26s/step - loss: 1.3136 - accuracy: 0.5026 - val_loss: 1.7424 - val_accuracy: 0.4688
Epoch 2/12
7/7 [==============================] - 156s 23s/step - loss: 1.2605 - accuracy: 0.5949 - val_loss: 1.6473 - val_accuracy: 0.5000
Epoch 3/12
7/7 [==============================] - 156s 23s/step - loss: 1.1407 - accuracy: 0.5179 - val_loss: 1.7023 - val_accuracy: 0.4922
Epoch 4/12
7/7 [==============================] - 157s 23s/step - loss: 1.0354 - accuracy: 0.5897 - val_loss: 1.3475 - val_accuracy: 0.5625
Epoch 5/12
7/7 [==============================] - 157s 24s/step - loss: 0.8201 - accuracy: 0.6718 - val_loss: 1.4221 - val_accuracy: 0.4844
Epoch 6/12
7/7 [==============================] - 156s 23s/step - loss: 0.8288 - accuracy: 0.6667 - val_loss: 1.3655 - val_accuracy: 0.5703
Epoch 7/12
7/7 [==============================] - 156s 23s/step - loss: 0.9217 - accuracy: 0.6718 - val_loss: 1.7844 - val_accuracy: 0.4922
Epoch 8/12
7/7 [==============================] - 157s 23s/step - loss: 0.9802 - accuracy: 0.6308 - val_loss: 1.4432 - val_accuracy: 0.4531
Epoch 9/12
7/7 [==============================] - 157s 23s/step - loss: 0.9497 - accuracy: 0.6513 - val_loss: 1.5416 - val_accuracy: 0.6016
Epoch 10/12
7/7 [==============================] - 157s 24s/step - loss: 0.6968 - accuracy: 0.7385 - val_loss: 1.4018 - val_accuracy: 0.5703
Epoch 11/12
7/7 [==============================] - 156s 23s/step - loss: 0.7696 - accuracy: 0.7282 - val_loss: 1.3748 - val_accuracy: 0.6094
Epoch 12/12
7/7 [==============================] - 157s 23s/step - loss: 0.6432 - accuracy: 0.7744 - val_loss: 1.3856 - val_accuracy: 0.5703
```


accuracy VS epochs
train
test
accuracy
epoch
0.75
0.70
0.65
0.60
0.55
0.50
0.45
0
2
4
6
8
10


loss vs epochs
train
test
accuracy
epoch
1.8
1.6
1.4
1.2
1.0
0.8
0.6
0
2
4
6
8
10

Do model checkpointing and callbacks help here?

Why did we use SoftMax Activation function here?

Converts a vector of numbers into a vector of probabilities

How can we improve this model?

- Imbalanced Classes
- Focusing on a recall-based approach
- More Images to train
- Retraining with Feedback

FLASK APPLICATION

Why Flask and not Node.js/Django?

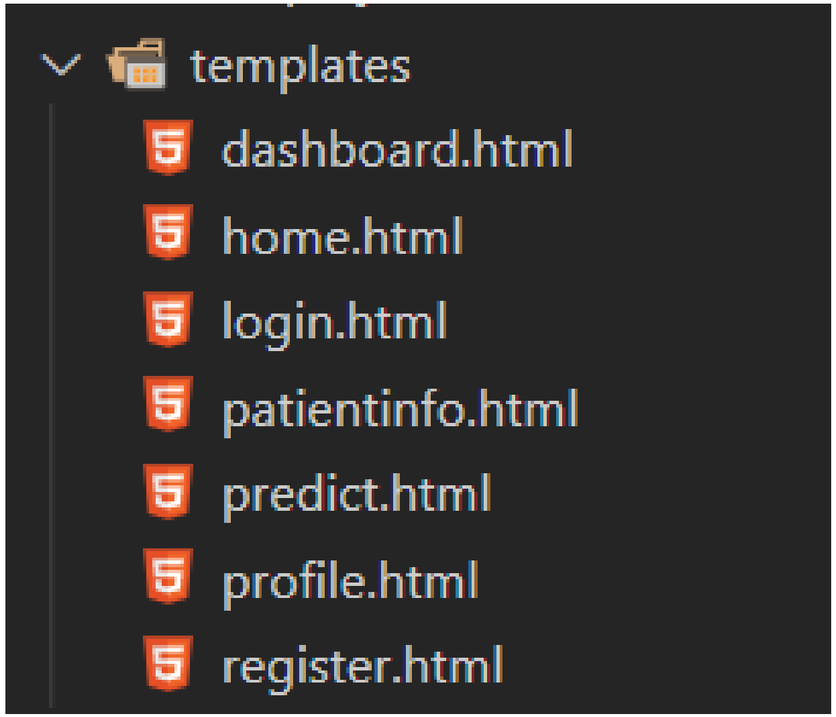
1. Light weight framework
2. Flexible to host on azure app service
3. Routing URL functions through Werkzeug makes the process easier.

Need for a DB

1. To make a platform that can support various other ML models that doctors could use
2. To keep track of a feedback table that can be used to retrain the model

Flask App Structure

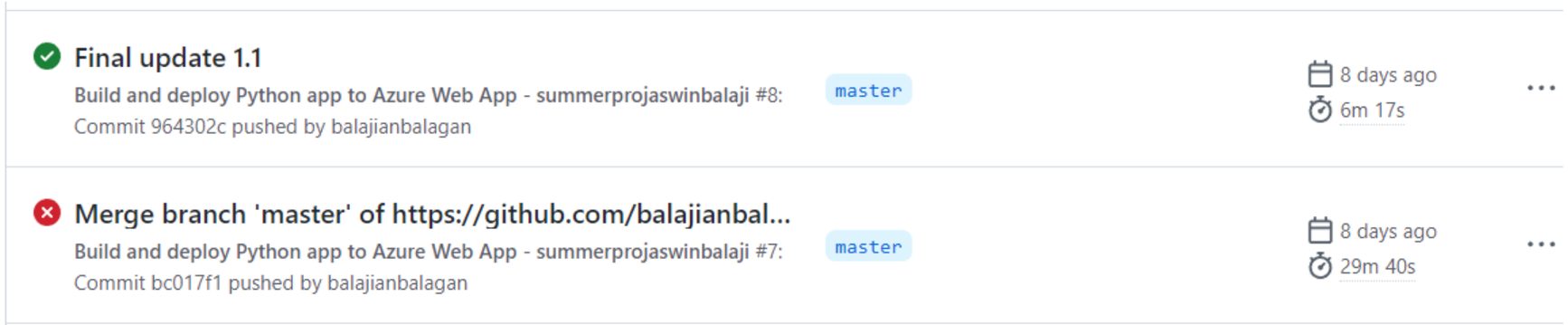
Front-end pages



Libraries used

```
import datetime
from distutils.log import debug
import os
import numpy as np
from flask import Flask
from flask import request
from flask import render_template
from flask import redirect, url_for, session
from flask_mysqldb import MySQL
from werkzeug.utils import secure_filename
import MySQLdb.cursors
import re
import random
import time
from dotenv import load_dotenv
from azure.storage.blob import BlobServiceClient
load_dotenv()
```

Failed builds



Lessons learnt

1. The need to containerize an application
2. Proper specification of requirements.txt

Database Structure

Table: accounts

Columns:

id	int AI PK
username	varchar(50)
password	varchar(255)
email	varchar(100)

Table: feedback

Columns:

sid	int AI PK
actualvalue	varchar(100)
predictedvalue	varchar(100)
iscorrect	tinyint(1)
imagelink	varchar(100)

Table: patient

Columns:

pid	int AI PK
did	int
startoftreatment	timestamp
contact	varchar(100)
pname	varchar(100)

Table: models

Columns:

mid	int AI PK
modelname	varchar(100)
noofparameters	int

Table: parameters

Columns:

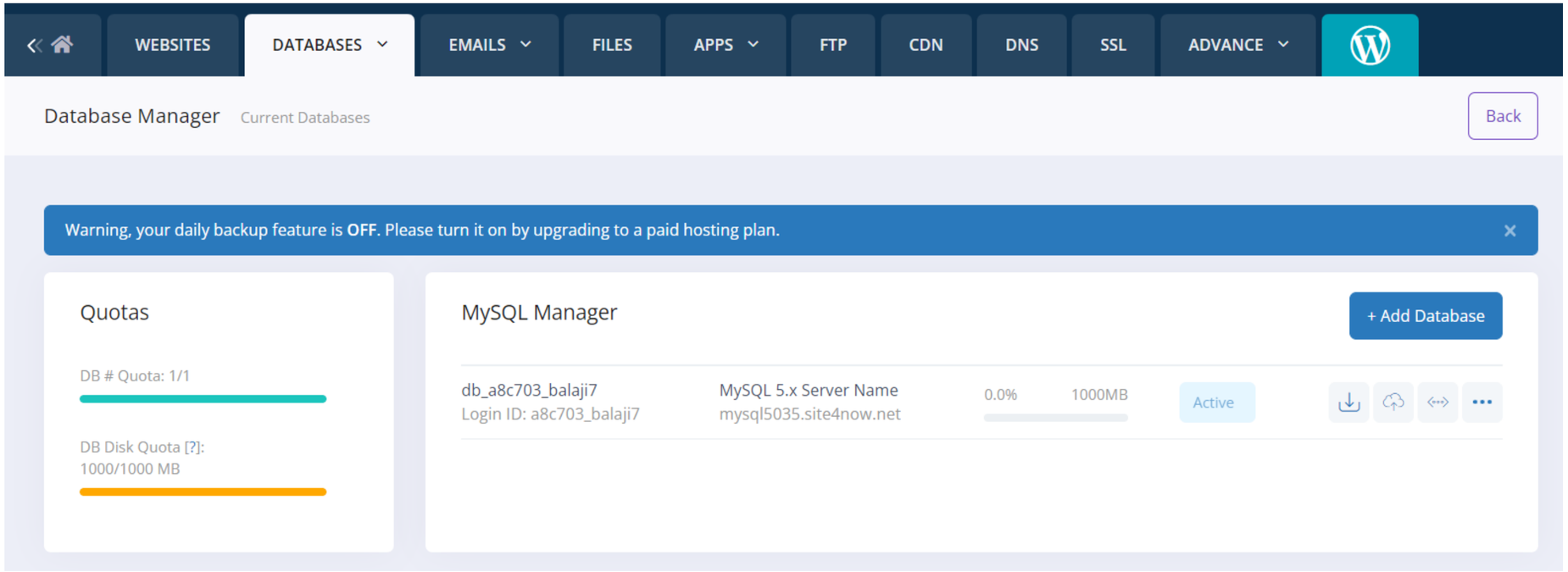
mid	int
paramname	varchar(100)
val	int

Table: submissions

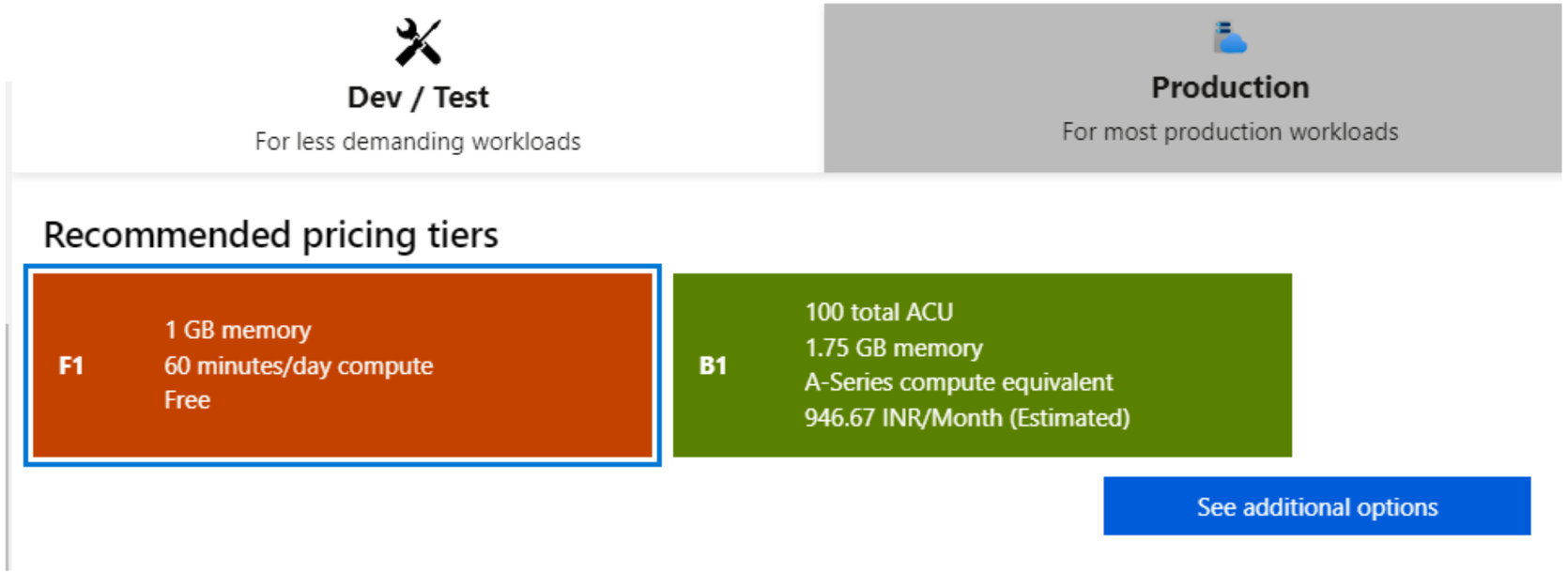
Columns:

sid	int AI PK
pid	int
mid	int
timeofsubmission	timestamp
result	varchar(100)
comments	varchar(100)
imagelink	varchar(100)

Cloud database service from smartasp.net



Azure app service



STUDIO SHODWE

Thank You