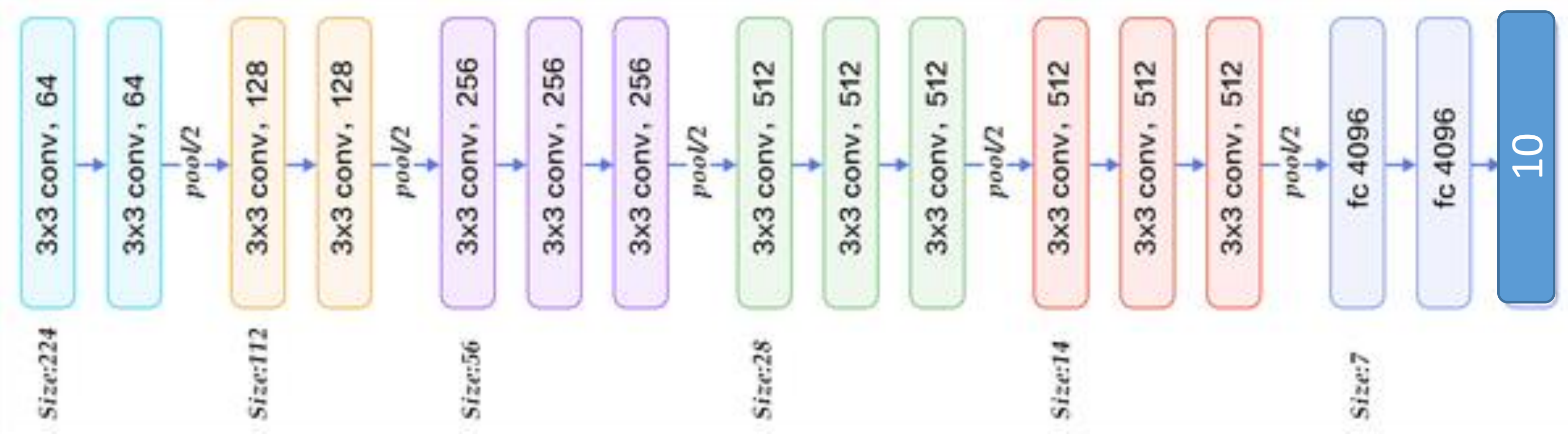
Zastosowanie sieci CNN uczonych na zaszumionych danych do klasyfikacji zdjęć

Architecture – VGG16

Original network had 3 FC layers at the end: 2 of those had 4096 neurons and the third last layer had as much layers as the number of categories.



My net has 4 FC layers at the end: 3 of those have 512 neurons and the last has 10 as the number of cities in the dataset.

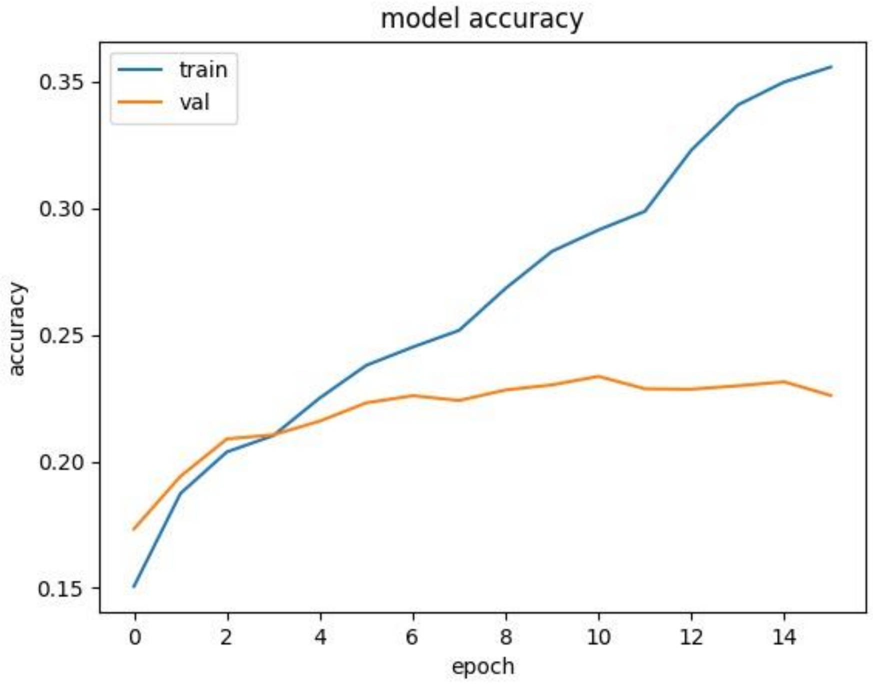
Architecture

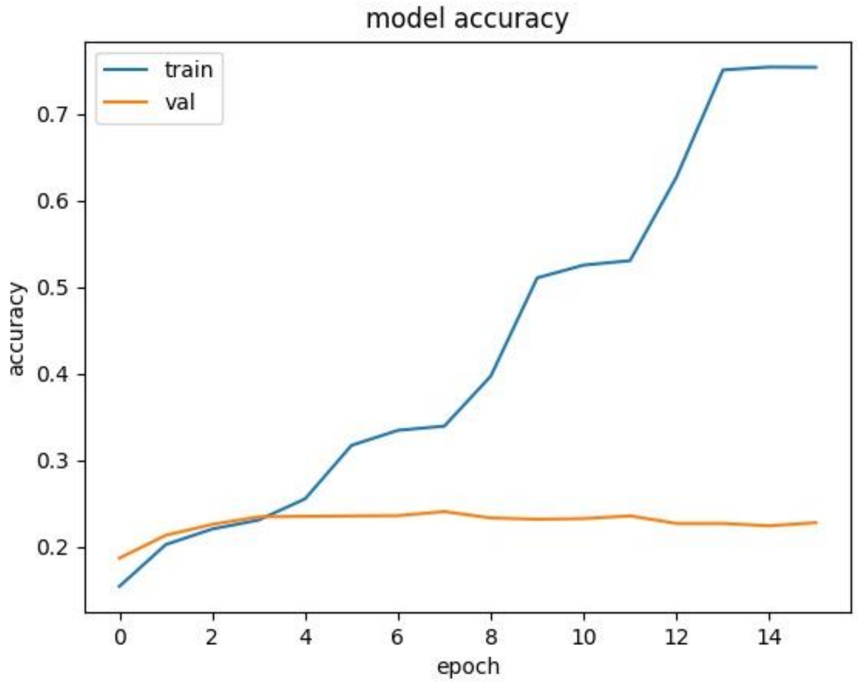
Modified:

- Fully connected layers: 512, 512, 512, 10
- Batch size: 4

Original:

- Fully connected layers: 4096, 4096, 10
- Batch size: 64

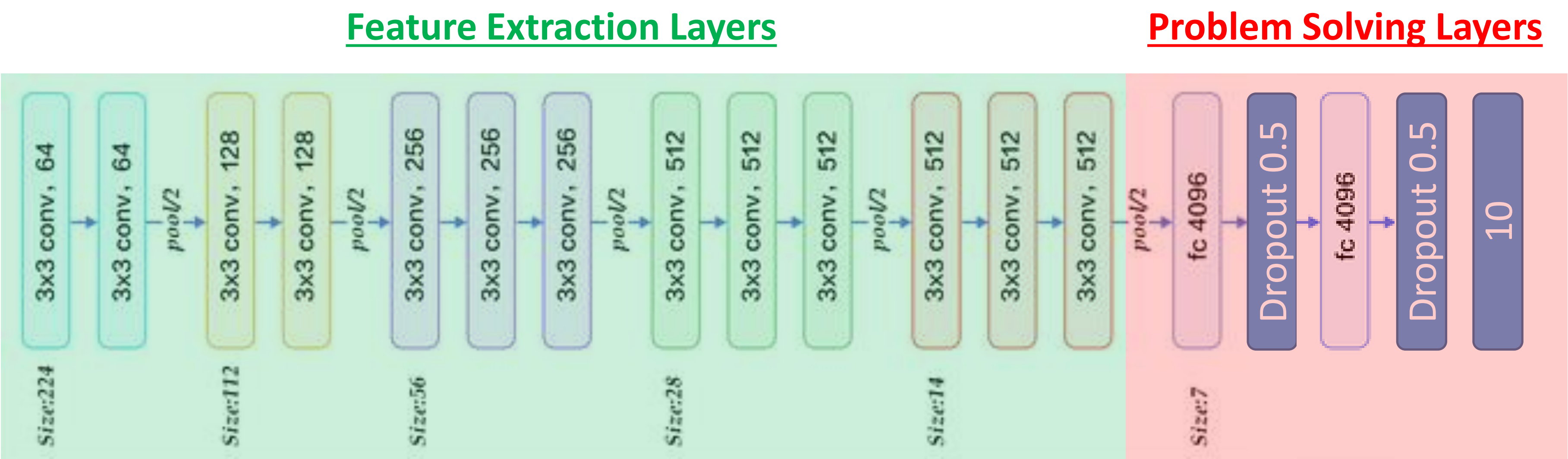




Architecture change

Original VGG16 was extended by adding regularization to enable longer training without overfitting. Regularization was introduced in the form of dropout after the fully connected layers with 4096 neurons. The magnitude of dropout was set to 0.5 which means at each iteration 50% of neurons were randomly dropped.

Training – procedure used previously

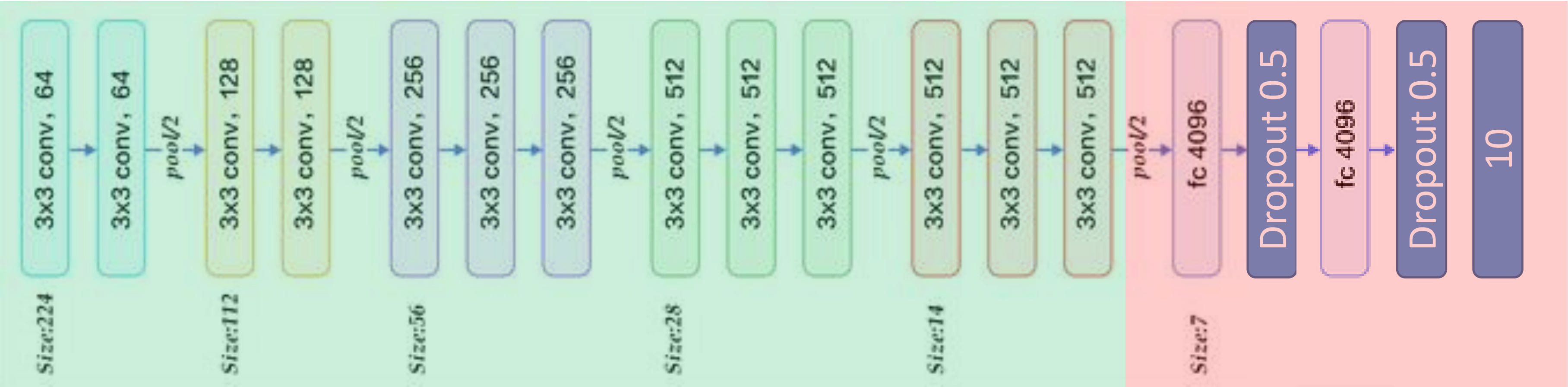


There are two main blocks of layers within CGG16 network: **feature extraction layers** which are responsible for extracting feature meaningful for the problem at hand and **problem solving layers** which are aimed at solving the problem.

Till now the training procedure used weights in **feature extraction layers** as the were (trained on ImageNet dataset containing 1k common objects) without changing them and only modified the **problem solving layers**. Training all weights was unsuccessful till now.

Training – procedure change

Feature Extraction Layers **Problem Solving Layers**



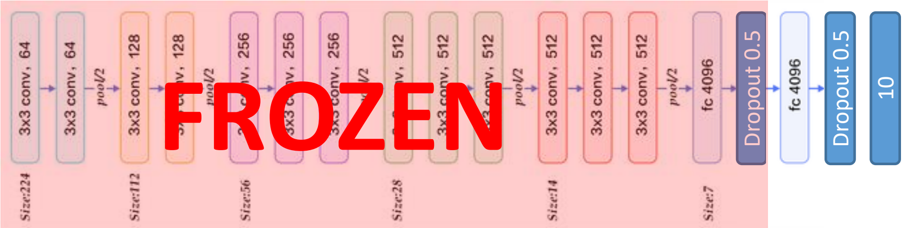
~~The current approach first trains the problem solving layers up to its maximum potential and once the validation error plateaus trains all layers (both feature extraction layers and problem solving layers) at a decreased learning rate. Providing a good solution as a starting point of the fine tuning process proves successful and the network is able to achieve a much higher results.~~

This was not correct: I froze all layers except for the last 3 ones (according to the original net design), but since I added regularization layers the last 3 layers were: SoftMax(10), Dropout(0.5) and Fully Connected(4096). This is not the entire problem solving group of layers.

Training – Comparison

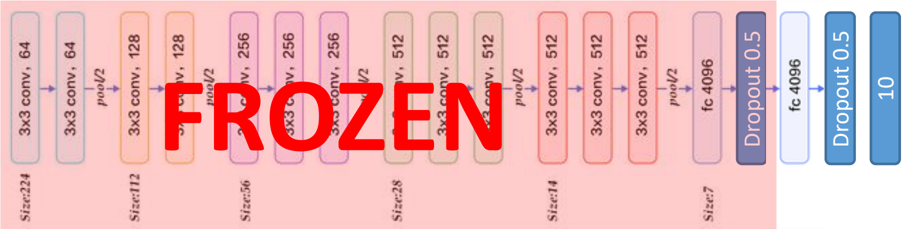
ADJUSTED slide.

Old approach:

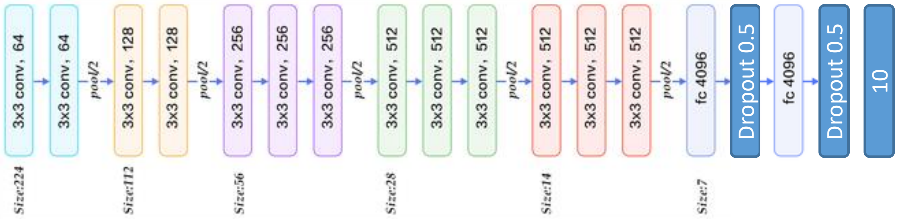


Feature extraction layers + 2 layers: FROZEN
Problem Solving Layers: TRAINABLE
Learning rate: 0.0001

New approach:



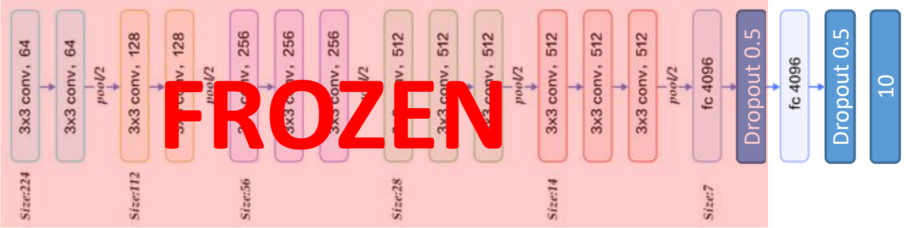
Feature extraction layers + 2 layers: FROZEN
Problem Solving Layers: TRAINABLE
Learning rate: 0.0001

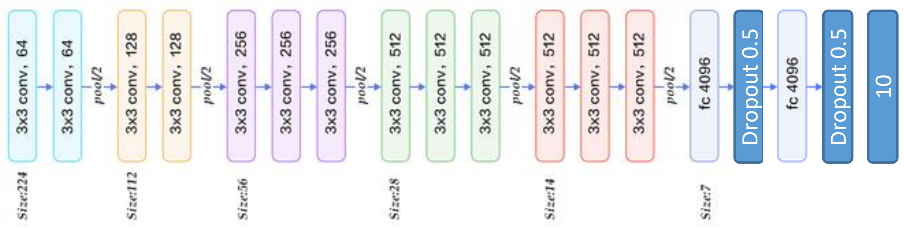


Feature extraction layers + 2 layers: TRAINABLE
Problem Solving Layers: TRAINABLE
Learning rate: 0.00001

Different ways of fine tuning comparison

First approach (originating in code error)

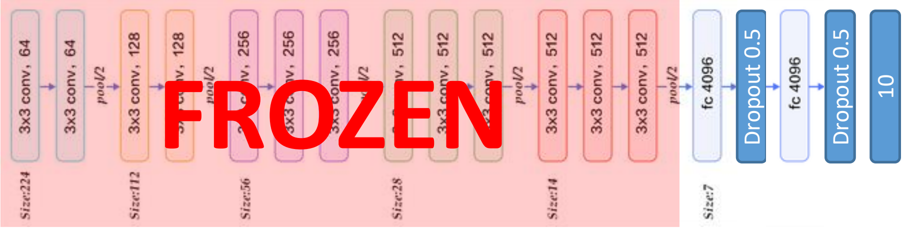


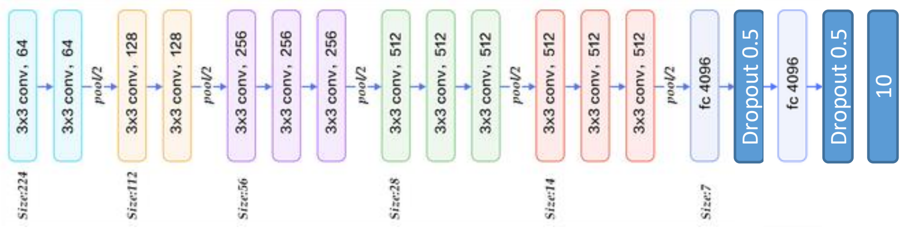


Feature extraction layers + 2 layers: FROZEN
Problem Solving Layers: TRAINABLE
Learning rate: 0.0001

Feature extraction layers + 2 layers: TRAINABLE
Problem Solving Layers: TRAINABLE
Learning rate: 0.00001

Second approach (aligned with theory)



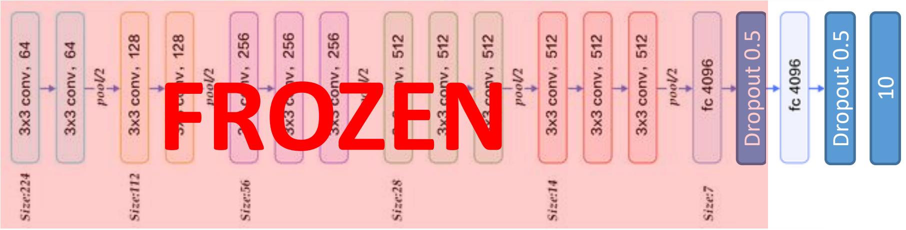


Feature extraction layers: FROZEN
Problem Solving Layers: TRAINABLE
Learning rate: 0.0001

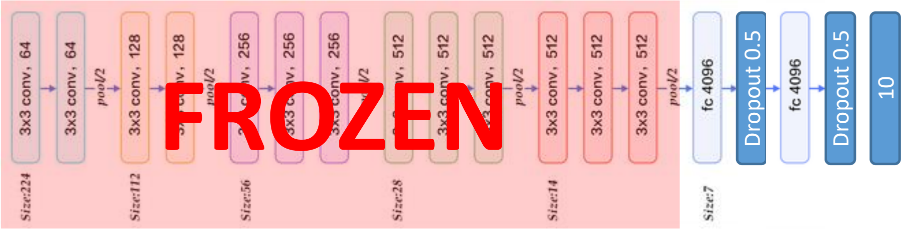
Feature extraction layers: TRAINABLE
Problem Solving Layers: TRAINABLE
Learning rate: 0.00001

Different ways of fine tuning comparison – results comparison

First approach (originating in code error)



Second approach (aligned with theory)



Experiment 1:

```
Random:
      Accuracy
mean  0.609867
std   0.007698
Test:
0.87
```

Experiment 2:

```
Random:
      Accuracy
mean  0.610033
std   0.005390
Test:
0.87
```

Experiment 3:

```
Random:
      Accuracy
mean  0.609433
std   0.005622
Test:
0.87
```

Experiment 1:

```
Random:
      Accuracy
mean  0.516567
std   0.006959
Test:
0.77
```

Experiment 2:

```
Random:
      Accuracy
mean  0.514300
std   0.007438
Test:
0.77
```

The results show that the error in the code lead to a much better result.

Food dataset

Food dataset - creation

List of top 10 foods:

- https://visual.ly/community/infographic/food/top-10-americas-favorite-foods
- https://food.ndtv.com/food-drinks/10-american-foods-777850
- http://islandgrownschools.weebly.com/uploads/1/0/7/8/10785576/top_ten_foods_consumed_in_america.pdf

Selected list of top 10 food is a mixture of the above sources to manage various restriction of training (popularity of hashtag on Instagram) and test (existence in food-101 data set) data availability. This list focuses more on America because the bias of Instagram.

1. Apple pie
2. Burger
3. Donuts
4. French Fries
5. Hot Dog
6. Macaroni and cheese
7. Pancake
8. Pizza
9. Spaghetti
10. Steak

Food dataset - creation

Data set split:

- Instagram Data – 800k images downloaded from Instagram containing one of the hashtags from the list of top 10 food. This data is divided into:
  - Training data – 770k images from 10 categories (equal number of images from each category)
  - Random testing data – 30k images from 10 categories (equal number of images from each category)
- Independent test data – 3k images from 10 categories (equal number of images from each category). This data comes from Kaggle and it was verified to contain one of the top 10 food.

Experiment hypothesis:

Once trained on noisy web data (not sure if class truly appears) we assume that the net will be able to categorize previously not seen NOT NOISY data with high accuracy. We want to validate the hypothesis by comparing results achieved for randomly selected datasets from Instagram that did not take part in the training procedure with independent test data where we know that the class appears.

Food dataset – is it characteristic?

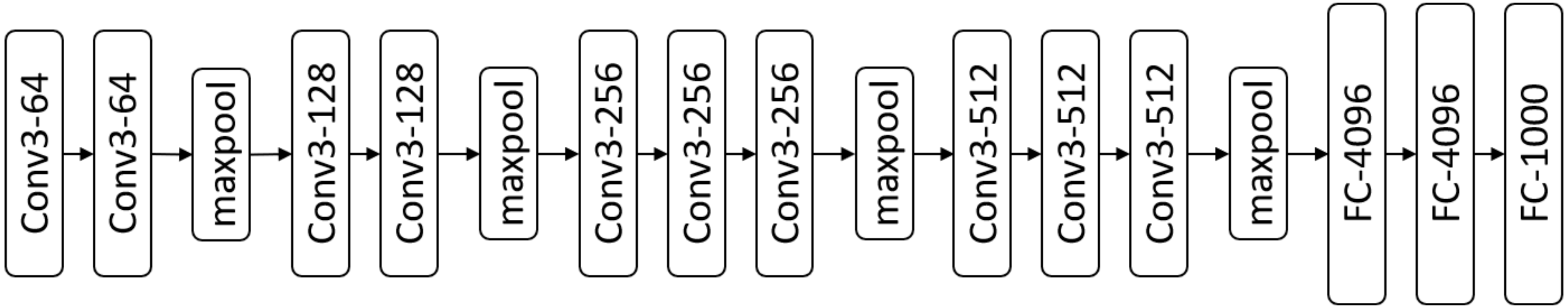
Food dataset – noisy data reminder

<u>Webly data</u>

There are various reasons why data associated with a particular hashtag might be incorrect:

- Label does not correspond to reality
- There are more than one class on the image
- The image is of low quality

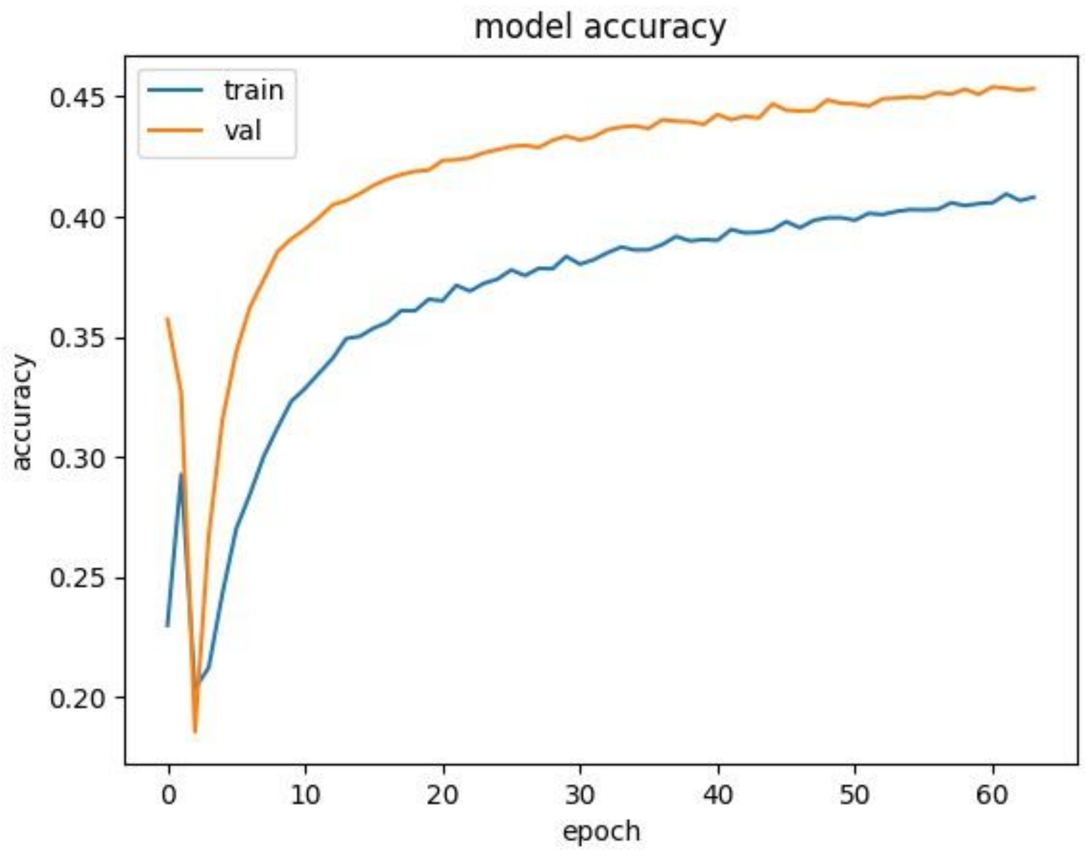
Below there are examples of the following categories: apple pie, burger and pancake.



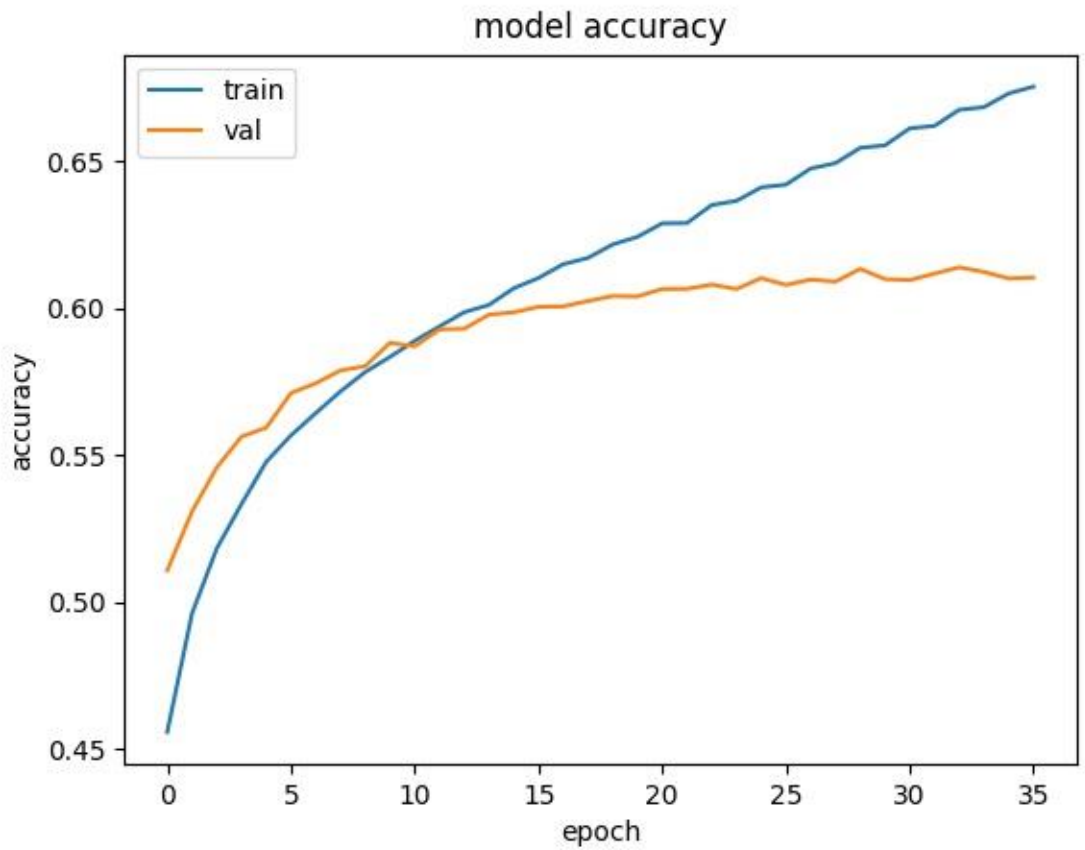
Food dataset – VGG16

Results

Problem solving training:



Fine tuning:



Results

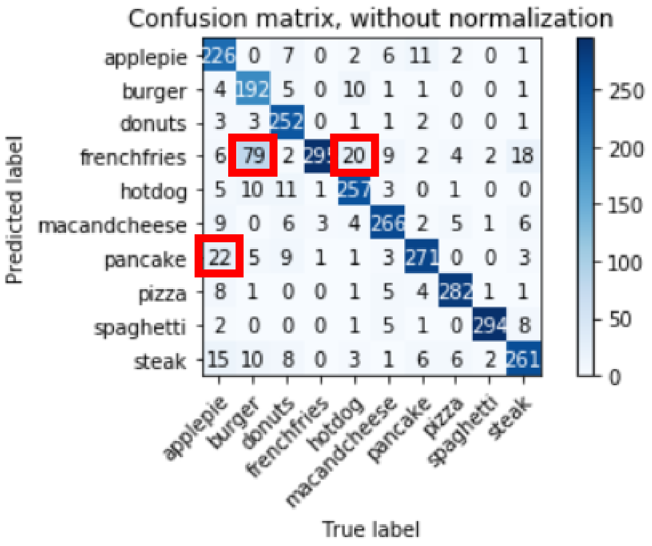
	Experiment 1:	Experiment 2:	Experiment 3:
Problem solving training:	Random: Accuracy mean 0.448800 std 0.007996 Test: 0.688	Random: Accuracy mean 0.457000 std 0.006609 Test: 0.7	Random: Accuracy mean 0.454900 std 0.005737 Test: 0.69
Fine tuning:	Random: Accuracy mean 0.609867 std 0.007698 Test: 0.87	Random: Accuracy mean 0.610033 std 0.005390 Test: 0.87	Random: Accuracy mean 0.609433 std 0.005622 Test: 0.87

Results – in-depth analysis

Fine tuning:

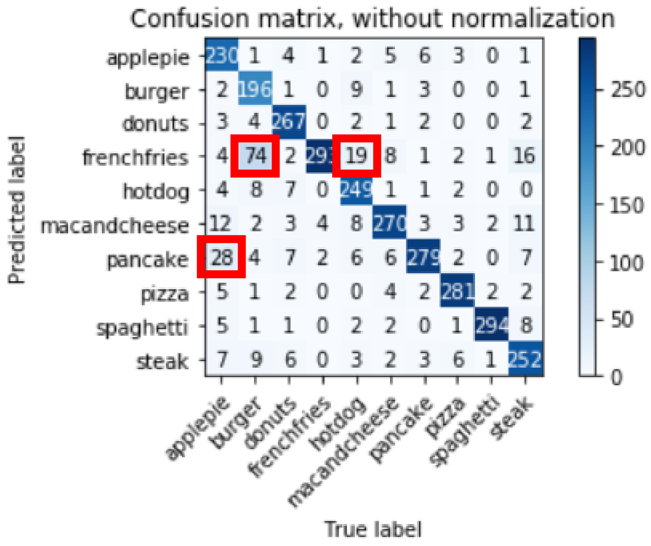
Experiment 1:

```
Test:
0.87
Test by vategory:
                  Accuracy
Categ
burger              0.64
applepie            0.75
donuts              0.84
hotdog              0.86
steak               0.87
macandcheese        0.89
pancake             0.90
pizza               0.94
spaghetti           0.98
frenchfries         0.98
Confusion matrix, without normalization
```



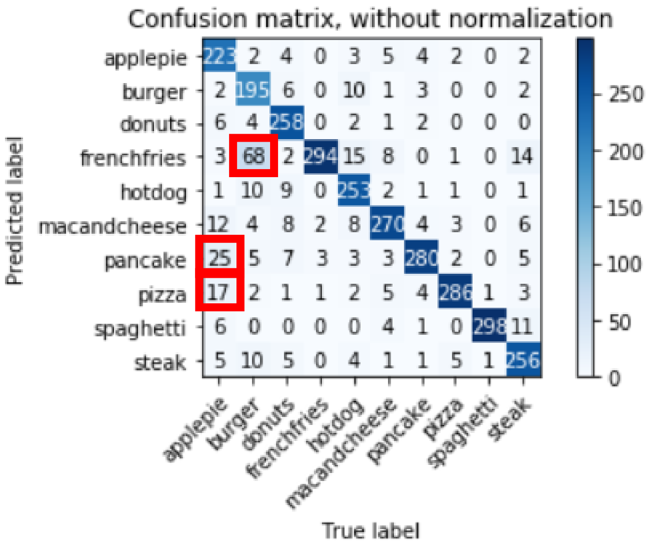
Experiment 2:

```
Test:
0.87
Test by vategory:
                  Accuracy
Categ
burger              0.65
applepie            0.77
hotdog              0.83
steak               0.84
donuts              0.89
macandcheese        0.90
pancake             0.93
pizza               0.94
frenchfries         0.98
spaghetti           0.98
Confusion matrix, without normalization
```



Experiment 3:

```
Test:
0.87
Test by vategory:
                  Accuracy
Categ
burger              0.65
applepie            0.74
hotdog              0.84
steak               0.85
donuts              0.86
macandcheese        0.90
pancake             0.93
pizza               0.95
frenchfries         0.98
spaghetti           0.99
Confusion matrix, without normalization
```



Results – in-depth analysis – Experiment 1

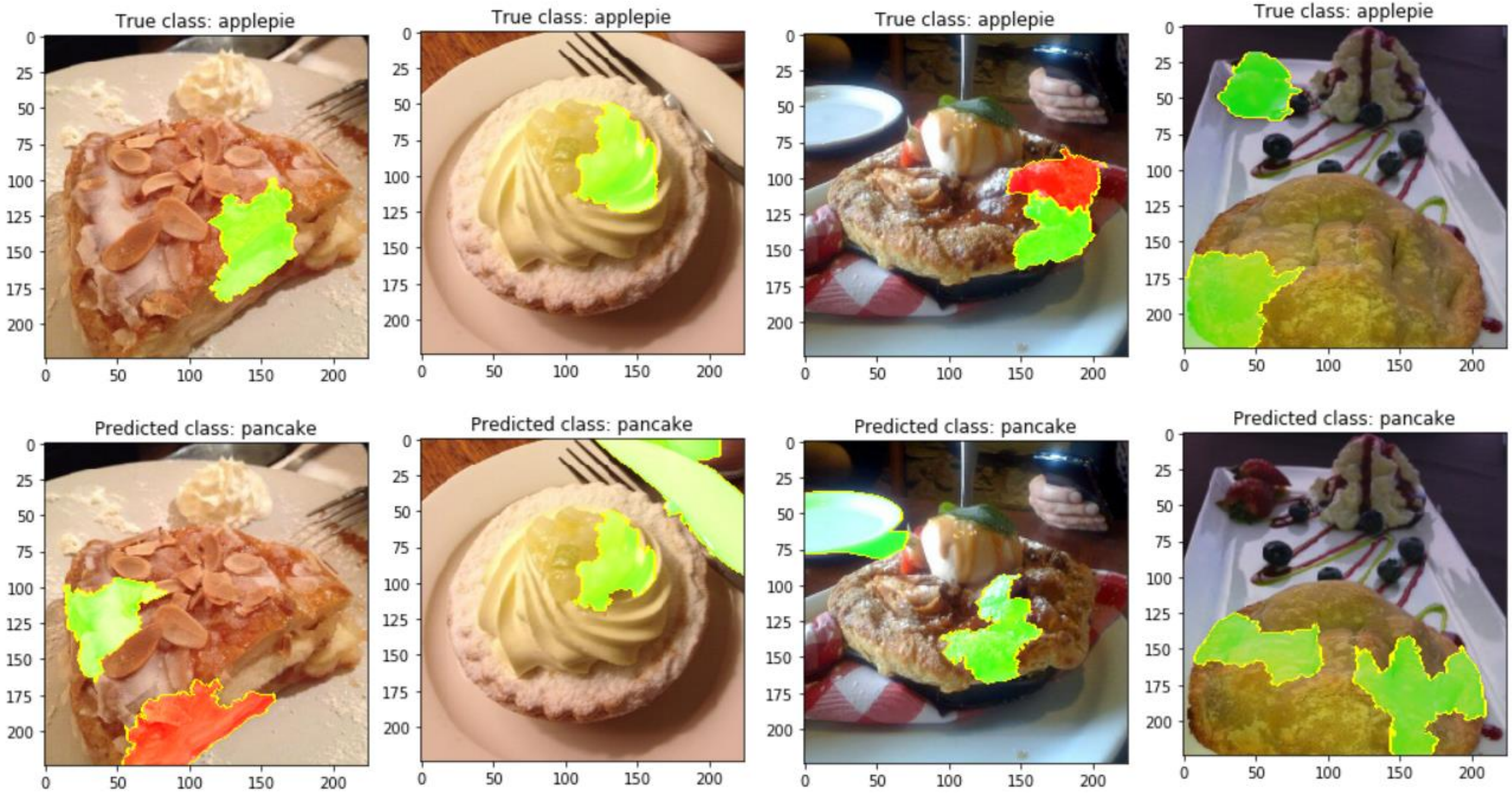
True class:

Predicted:

Results – in-depth analysis – Experiment 1

True class:

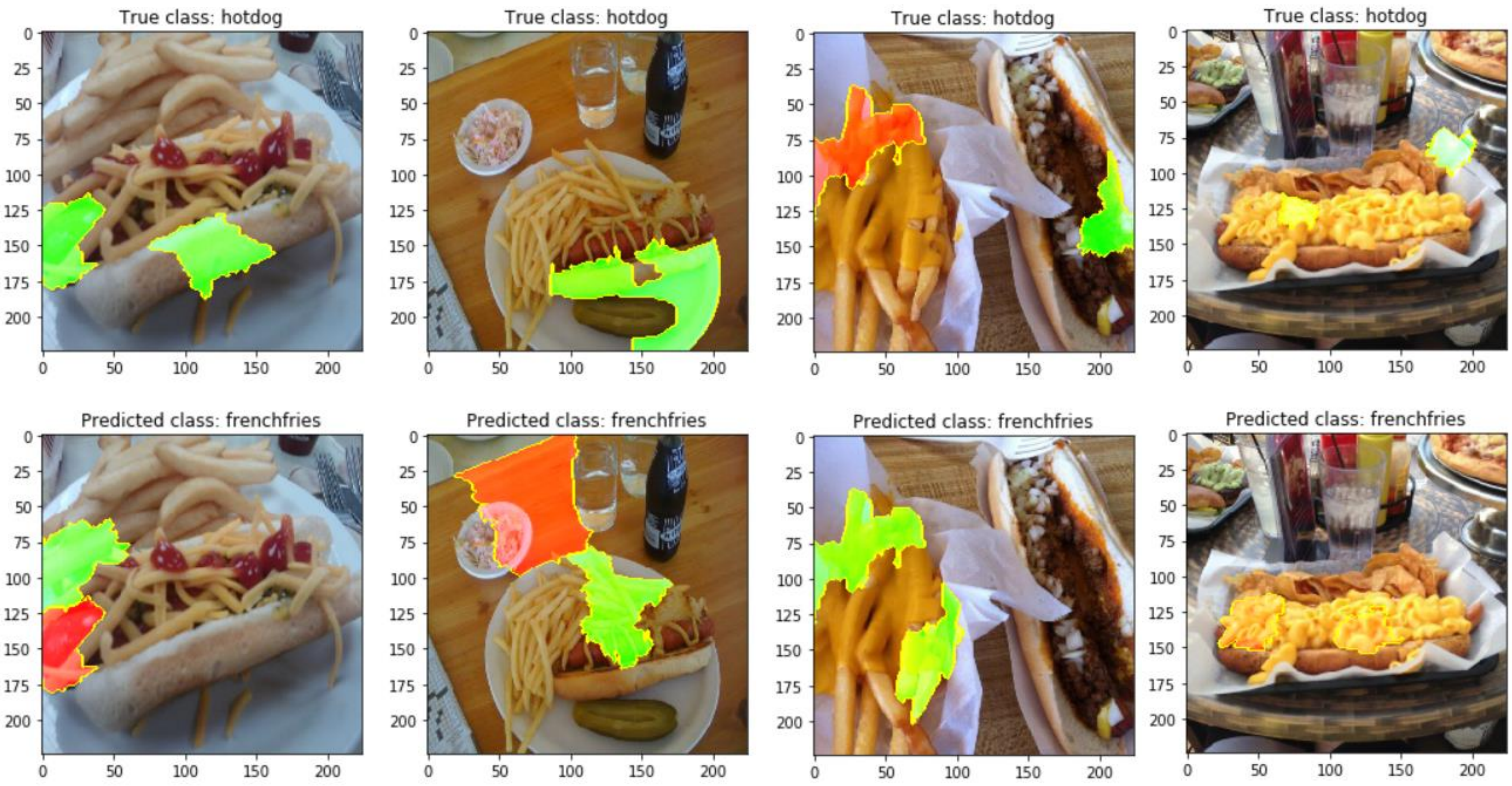
Predicted:



Results – in-depth analysis – Experiment 1

True class:

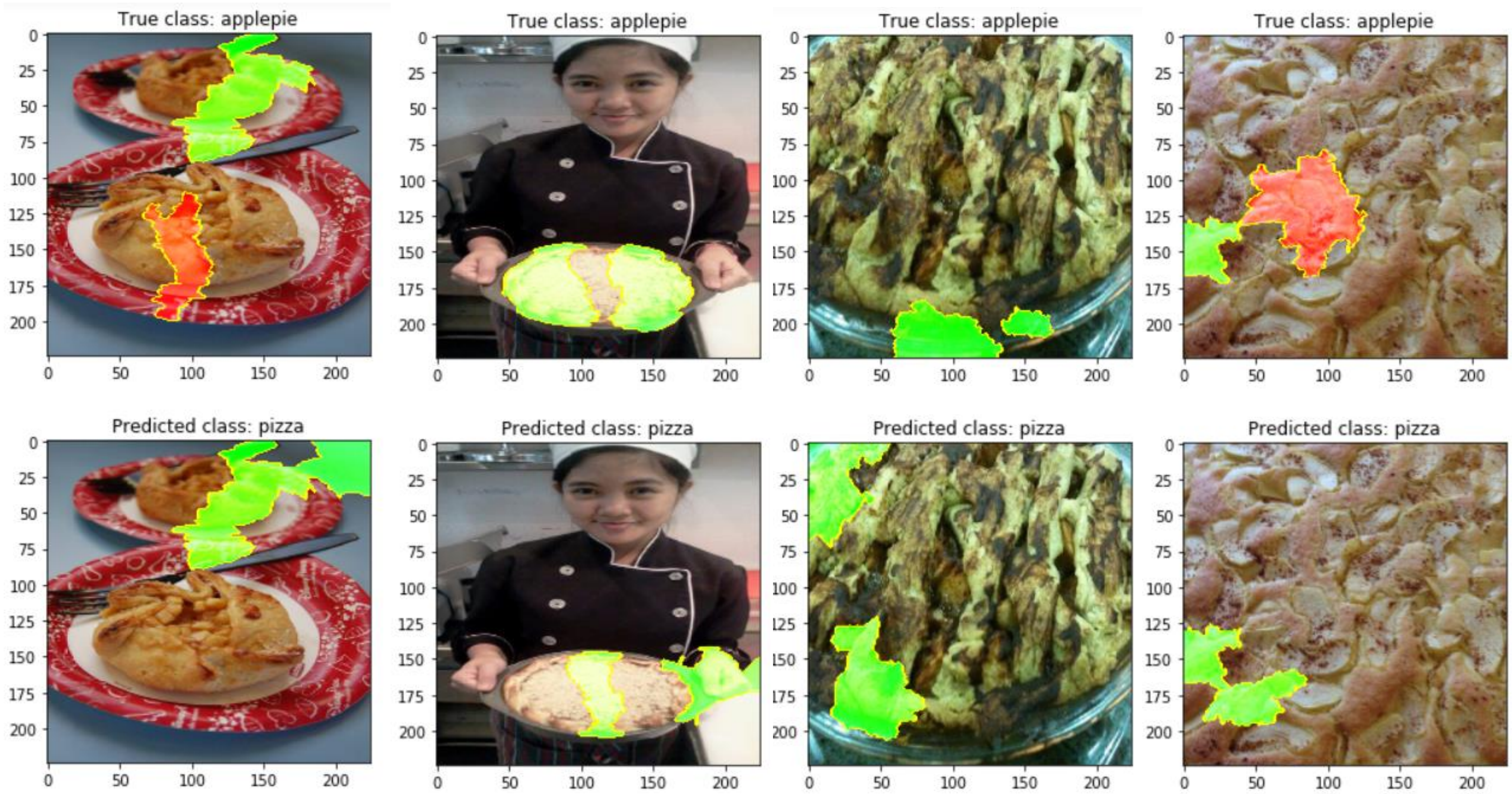
Predicted:

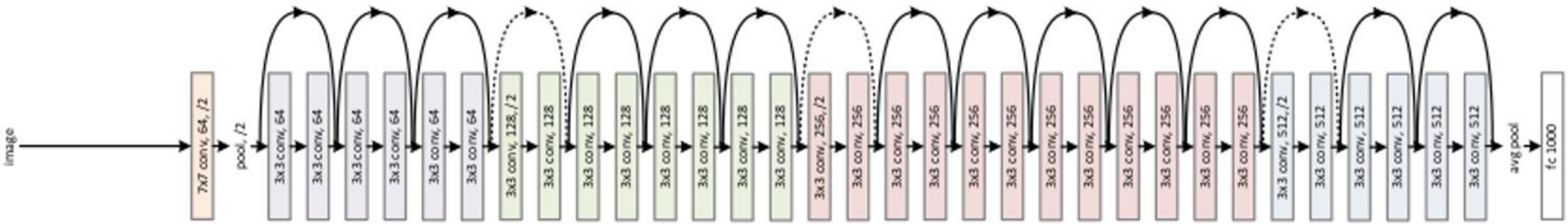


Results – in-depth analysis – Experiment 3

True class:

Predicted:

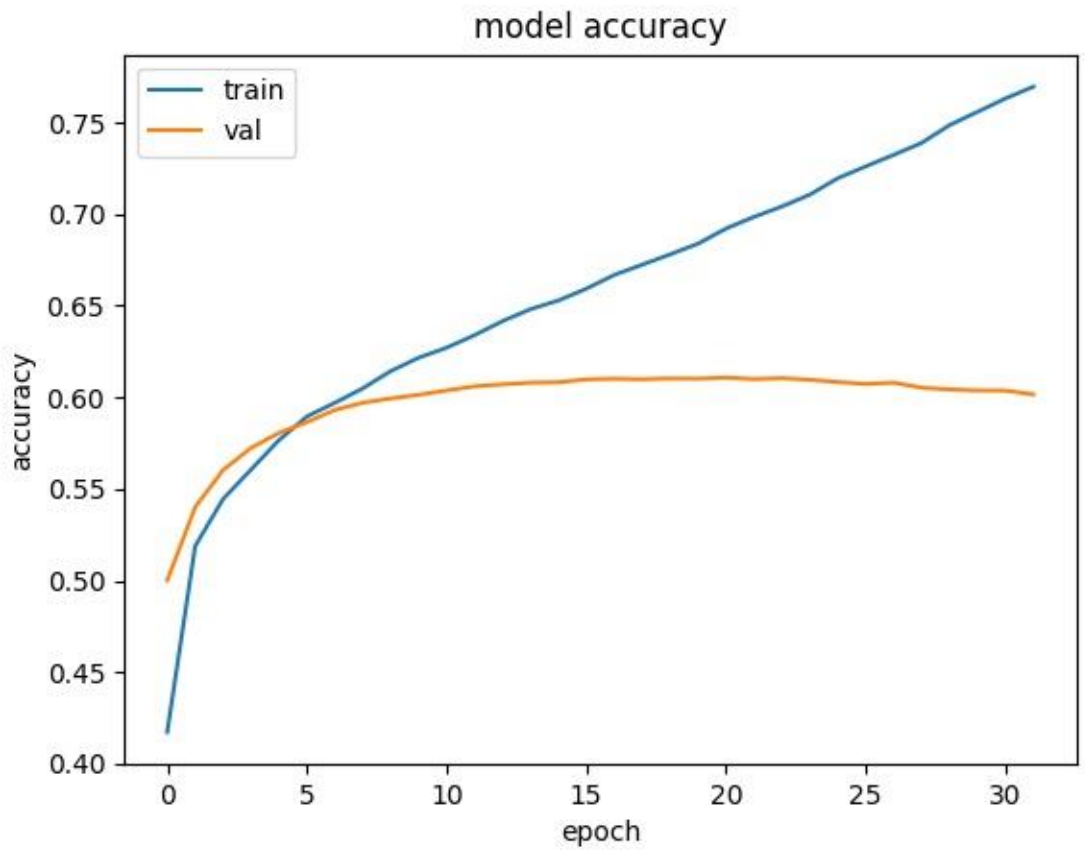




Food dataset – ResNet

Results

Fine tuning:



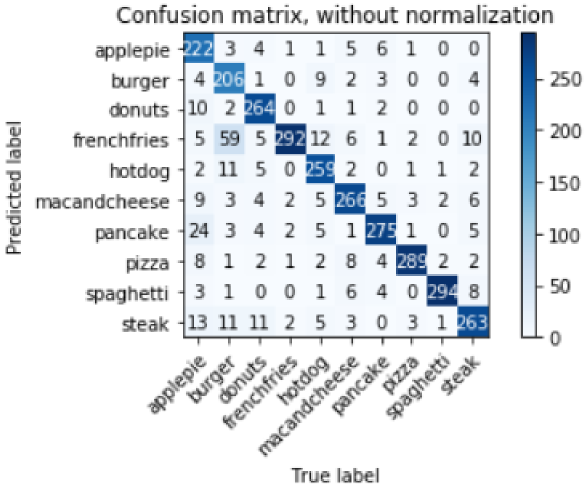
ResNet was able to achieve the same level of accuracy in a shorter time span. It did not require a 2 stage training process (problem solving layers training, fine tuning). This result was achieved training all neurons since the beginning at a learning rate of 0.00001 (the same as for fine tuning in VGG16).

Results – in-depth analysis

Fine tuning:

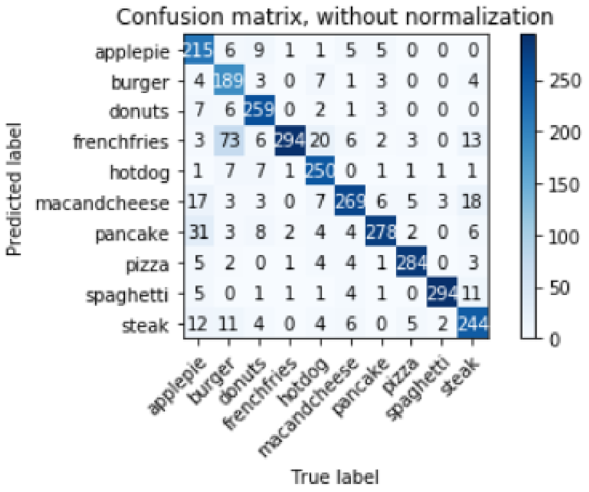
Experiment 1:

```
Random:
           Accuracy
mean   0.604800
std    0.007096
Test:
0.88
Test by vategory:
                  Accuracy
Categ
burger              0.69
applepie            0.74
hotdog              0.86
steak               0.88
donuts              0.88
macandcheese        0.89
pancake             0.92
pizza               0.96
frenchfries         0.97
spaghetti           0.98
Confusion matrix, without normalization
```



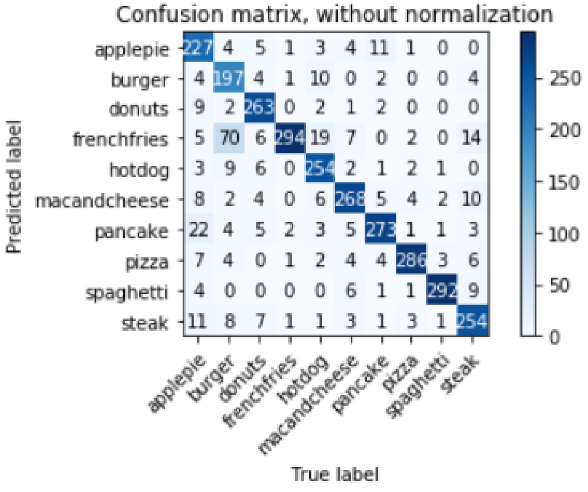
Experiment 2:

```
Random:
           Accuracy
mean   0.605133
std    0.007962
Test:
0.86
Test by vategory:
                  Accuracy
Categ
burger              0.63
applepie            0.72
steak               0.81
hotdog              0.83
donuts              0.86
macandcheese        0.90
pancake             0.93
pizza               0.95
frenchfries         0.98
spaghetti           0.98
Confusion matrix, without normalization
```



Experiment 3:

```
Random:
           Accuracy
mean   0.605867
std    0.006317
Test:
0.87
Test by vategory:
                  Accuracy
Categ
burger              0.66
applepie            0.76
hotdog              0.85
steak               0.85
donuts              0.88
macandcheese        0.89
pancake             0.91
pizza               0.95
spaghetti           0.97
frenchfries         0.98
Confusion matrix, without normalization
```



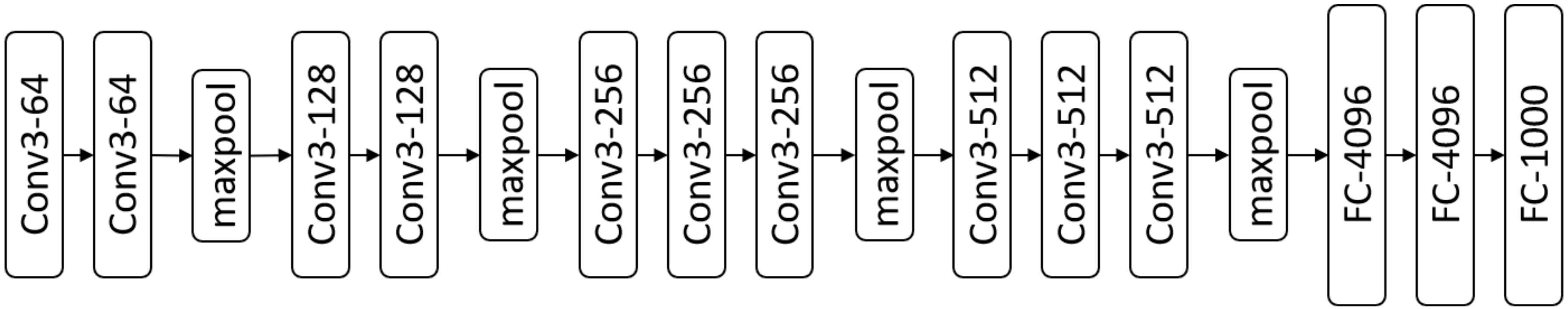
Instacities dataset

Food dataset - creation

Data set split:

- Instagram Data – 800k images downloaded from Instagram containing one of the hashtags from the list of 10 cities. This data is divided into:
  - Training data – 770k images from 10 categories (equal number of images from each category)
  - Random testing data – 30k images from 10 categories (equal number of images from each category)
- Independent test data – this data comes from official Instagram accounts of the cities in training set. The list of accounts is presented below. Each account has a various number of images. We have constructed 2 test sets from those images one of random 300 images per category and the other with images that we believe are characteristic for the city (like “Big Ben” for London).
  - @chicago
  - @cityofmelbourne
  - @london
  - @losangeles_city
  - @nycgov
  - @onlyinsf
  - @seetorontonow
  - @sydney
  - @visit_singapore

Food dataset – is it characteristic?



Instacities dataset – VGG16

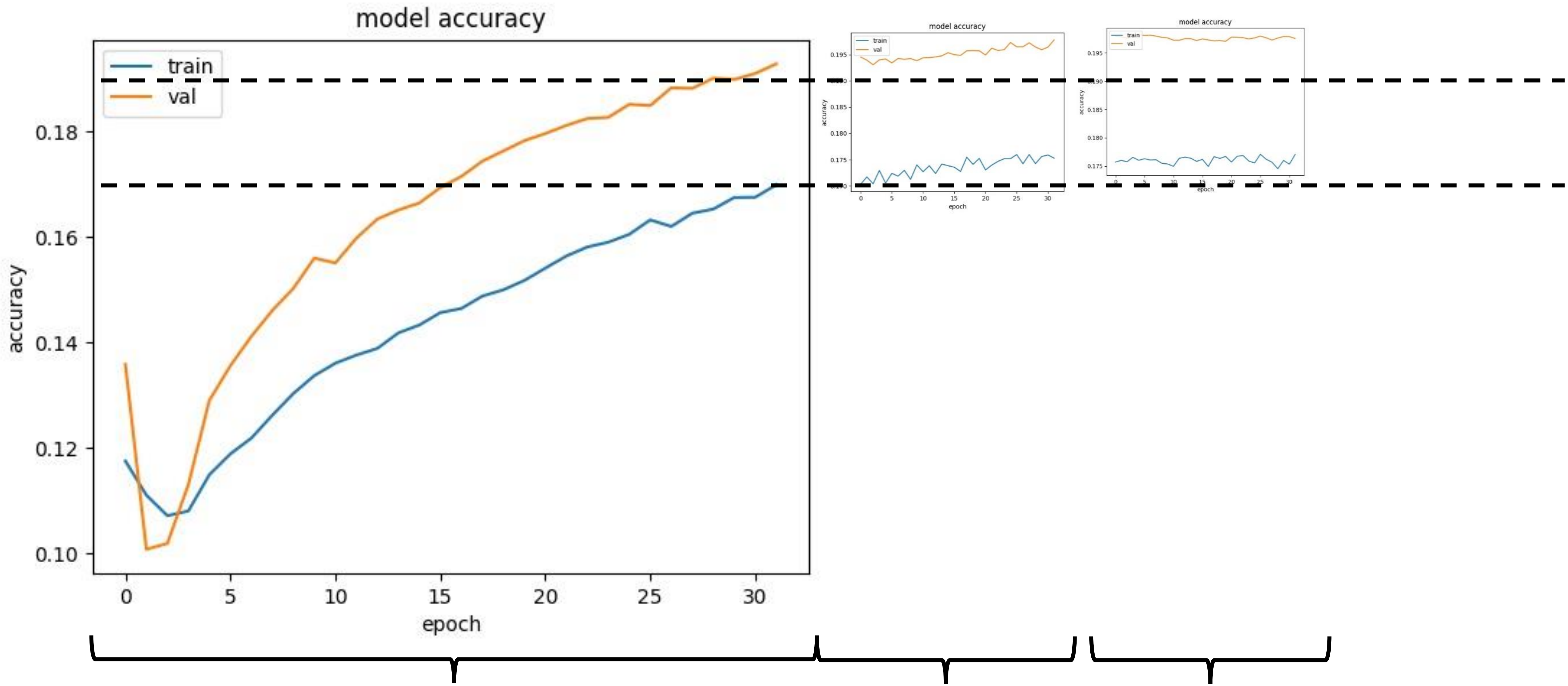
Results

The process for training VGG16 net for Instacities dataset was a bit more complex. Eventually I used a setup with 5 stages but it could probably be reduced to 4 or less.

Stage	Trainable layers	Learning Rate
Stage 1	Last 3	1e-4
Stage 2	Last 3	1e-5
Stage 3	Last 3	1e-6
Stage 4	All	1e-5
Stage 5	All	1e-6

The majority of knowledge extraction and the biggest improvement can be seen in stages 1 and 4 which initiate learning some of the layers.

Results

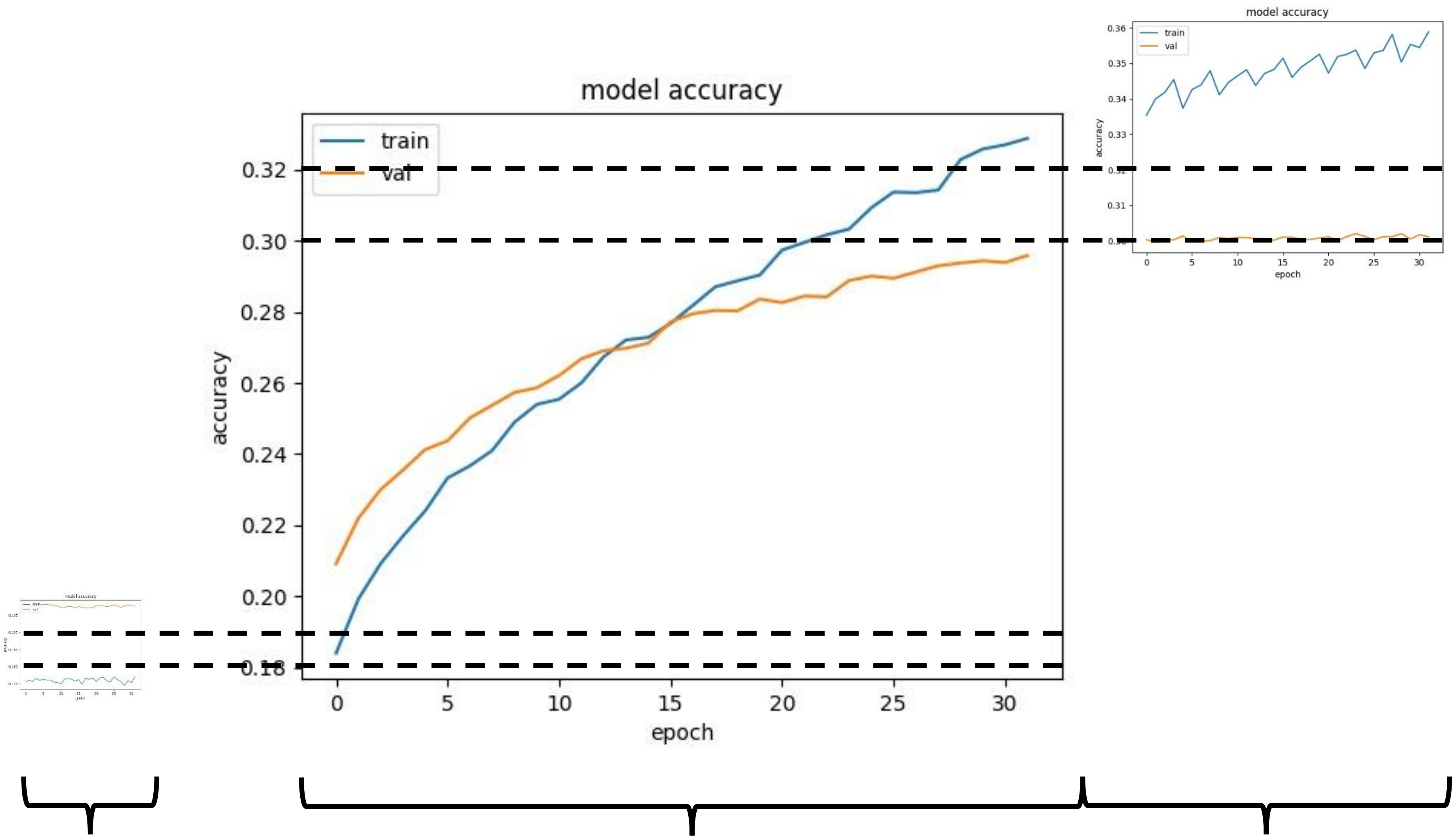


Stage 1

Stage 2

Stage 3

Results



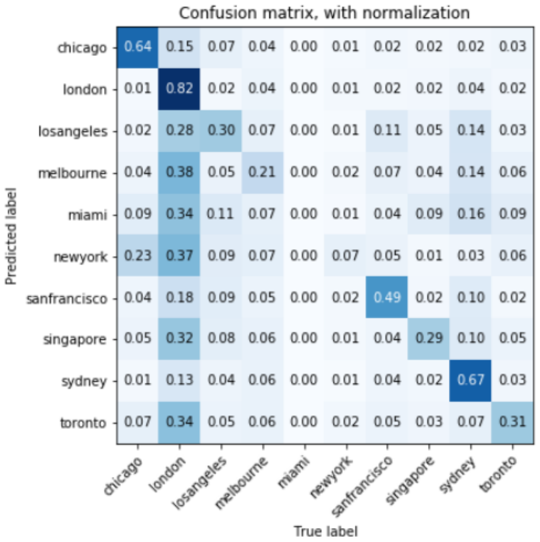
Stage 3 **Stage 4** **Stage 5**

Results – in-depth analysis – all test cases

Fine tuning:

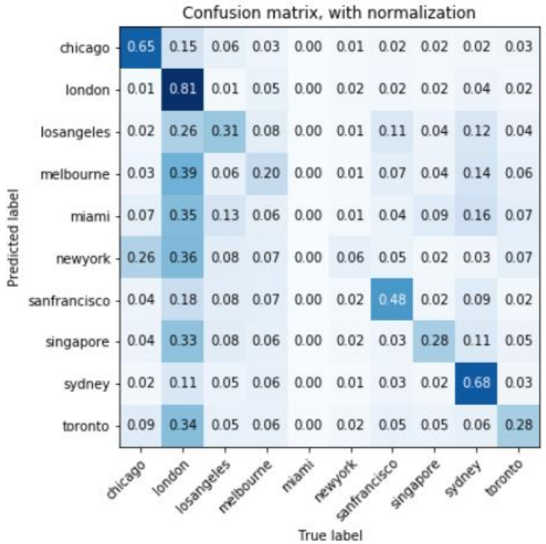
Experiment 1:

```
Random:
          Accuracy
mean   0.303167
std    0.007492
Test:
0.45
Test by vategory:
                 Accuracy
Categ
toronto              0.26
losangeles           0.28
newyork              0.38
melbourne            0.39
sanfrancisco         0.42
london               0.46
singapore            0.47
sydney               0.50
chicago              0.62
```



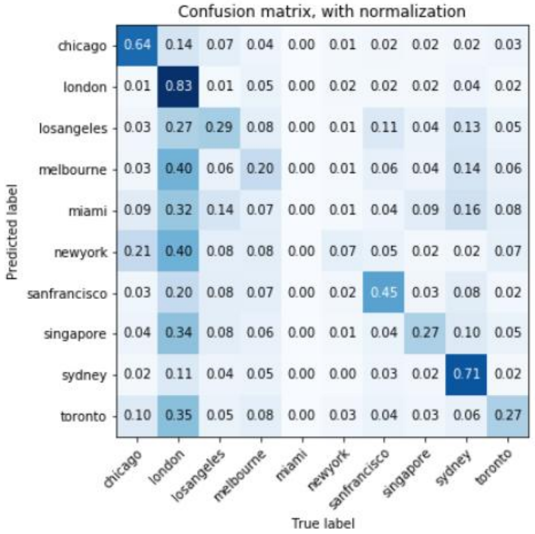
Experiment 2:

```
Random:
          Accuracy
mean   0.300500
std    0.006717
Test:
0.45
Test by vategory:
                 Accuracy
Categ
toronto              0.26
losangeles           0.28
melbourne            0.34
newyork              0.35
sanfrancisco         0.40
london               0.47
singapore            0.51
sydney               0.52
chicago              0.60
```



Experiment 3:

```
Random:
          Accuracy
mean   0.300767
std    0.008439
Test:
0.46
Test by vategory:
                 Accuracy
Categ
toronto              0.28
losangeles           0.31
melbourne            0.32
newyork              0.35
sanfrancisco         0.42
london               0.46
singapore            0.53
sydney               0.53
chicago              0.65
```



Results – in-depth analysis – 300 from test

Experiment 1:

10 x Randomly selected 300 from test images:

```
Test Random:
0.42 (+/-0.0073)
Test Random by category MEAN:
chicago          0.62
london           0.46
losangeles       0.28
melbourne        0.39
newyork          0.38
sanfrancisco     0.41
singapore        0.47
sydney           0.50
toronto          0.25
dtype: float64
Test Random by category STD:
chicago          0.0276
london           0.0325
losangeles       0.0219
melbourne        0.0237
newyork          0.0144
sanfrancisco     0.0269
singapore        0.0242
sydney           0.0266
toronto          0.0238
dtype: float64
```

Selected 300 from test images:

```
Test Selected:
0.71
Test Selected by vategory:
              Accuracy
Categ
chicago           0.58
london            0.59
sydney            0.69
losangeles        0.72
melbourne         0.78
sanfrancisco      0.80
toronto           0.84
newyork           0.86
singapore         0.86
```

Experiment 2:

```
Test Random:
0.42 (+/-0.0079)
Test Random by category MEAN:
chicago          0.61
london           0.47
losangeles       0.28
melbourne        0.34
newyork          0.35
sanfrancisco     0.41
singapore        0.51
sydney           0.52
toronto          0.26
dtype: float64
Test Random by category STD:
chicago          0.0279
london           0.0280
losangeles       0.0228
melbourne        0.0219
newyork          0.0146
sanfrancisco     0.0273
singapore        0.0238
sydney           0.0285
toronto          0.0217
dtype: float64
Test Selected:
0.69
Test Selected by vategory:
              Accuracy
Categ
london            0.57
chicago           0.58
losangeles        0.63
sydney            0.70
melbourne         0.73
newyork           0.77
sanfrancisco      0.79
toronto           0.80
singapore         0.87
```

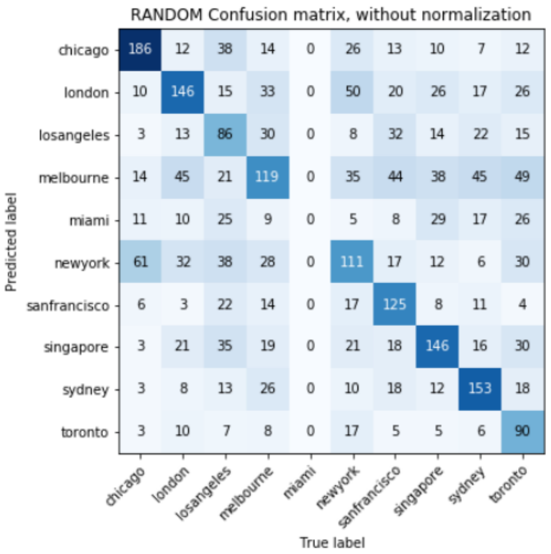
Experiment 3:

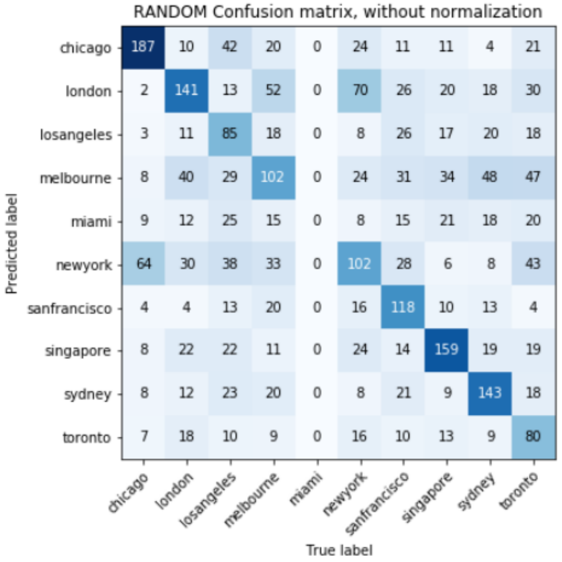
```
Test Random:
0.43 (+/-0.0092)
Test Random by category MEAN:
chicago          0.65
london           0.46
losangeles       0.30
melbourne        0.32
newyork          0.35
sanfrancisco     0.43
singapore        0.53
sydney           0.53
toronto          0.28
dtype: float64
Test Random by category STD:
chicago          0.0240
london           0.0292
losangeles       0.0246
melbourne        0.0241
newyork          0.0151
sanfrancisco     0.0253
singapore        0.0253
sydney           0.0280
toronto          0.0218
dtype: float64
Test Selected:
0.7
Test Selected by vategory:
              Accuracy
Categ
chicago           0.58
london            0.58
losangeles        0.65
melbourne         0.69
sydney            0.71
newyork           0.77
toronto           0.81
sanfrancisco      0.83
singapore         0.89
```

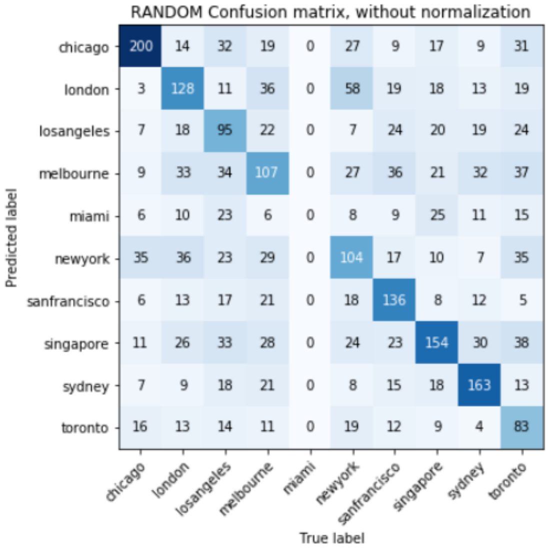
Results – in-depth analysis – 300 from test

Experiment 1: **Experiment 2:** **Experiment 3:**

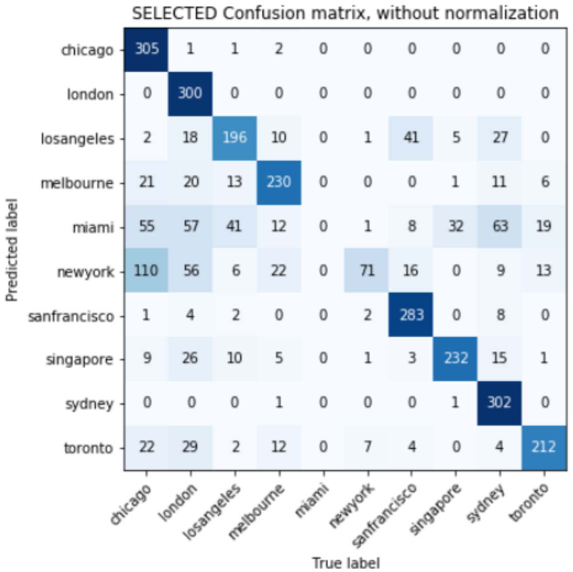
10 x Randomly selected 300 from test images:

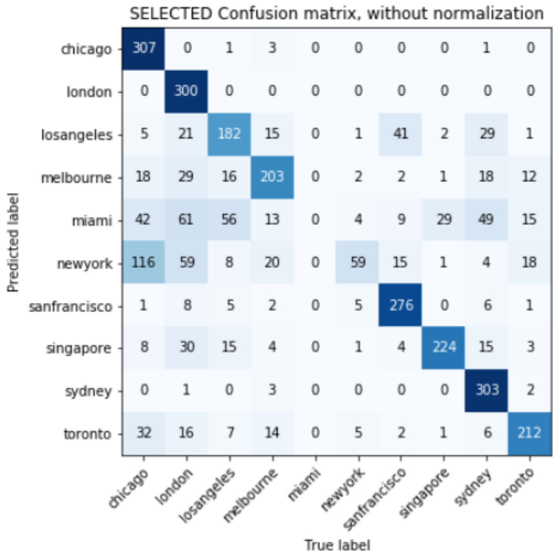


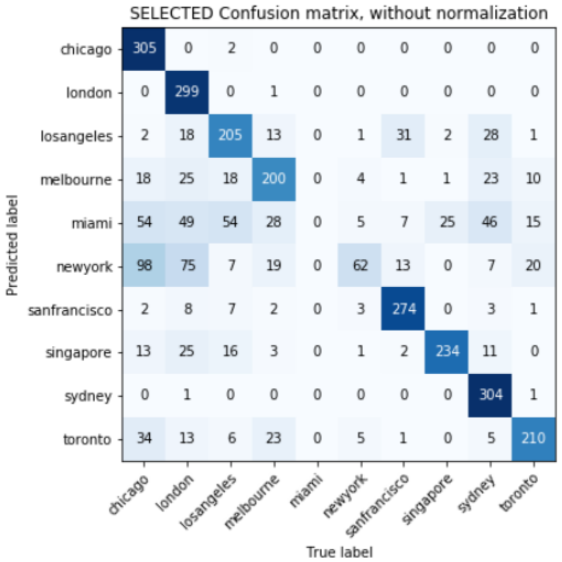


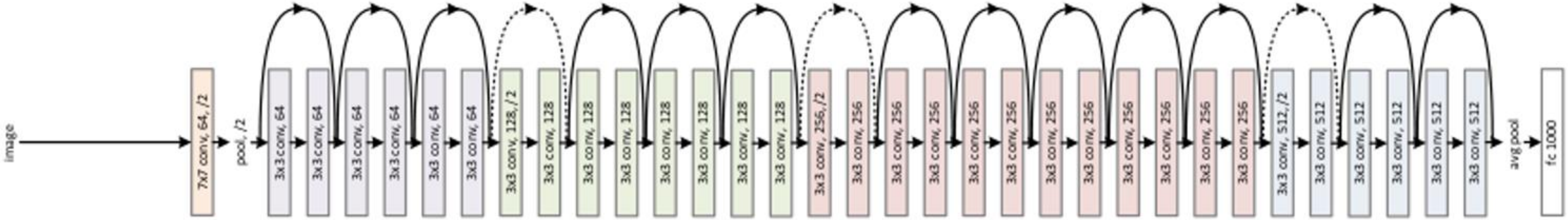


Selected 300 from test images:





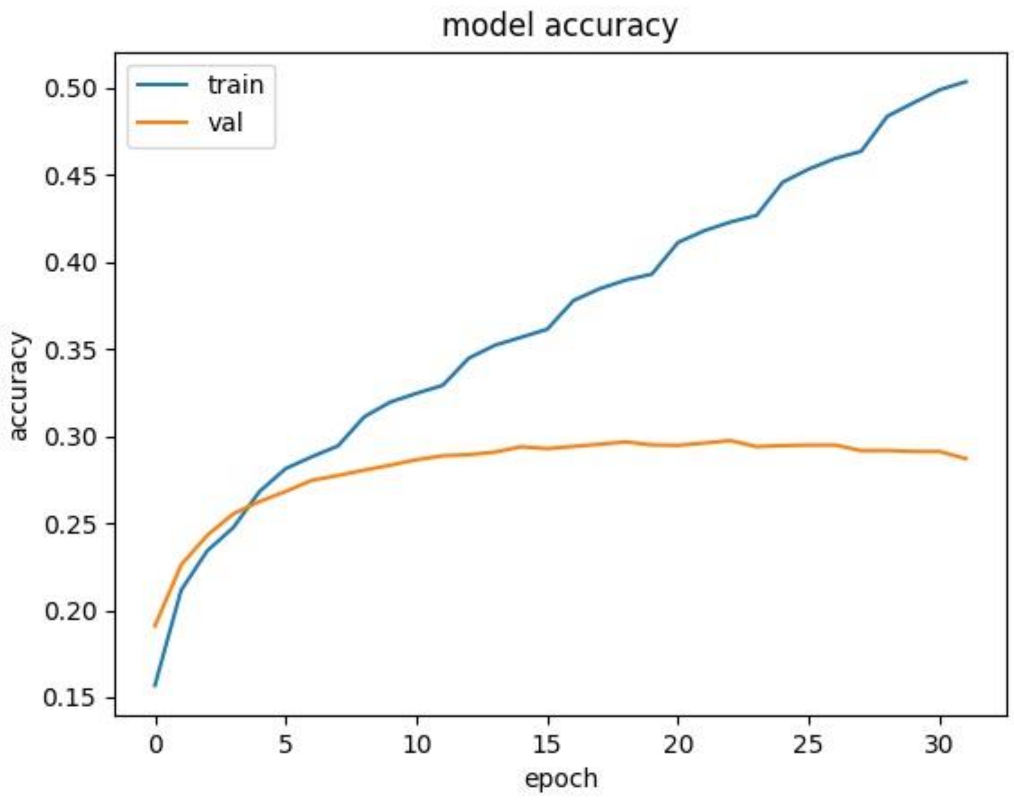




Instacities dataset – ResNet

Results

Fine tuning:



ResNet was able to achieve similar level of accuracy in a shorter time span. It did not require a 5 stage training process (problem solving layers training x3, fine tuning x2).

Results – in-depth analysis

10 x Randomly selected 300 from test images:

Selected 300 from test images:

Experiment 1:

```
Test Random:
0.4 (+/-0.0242)
Test Random by category MEAN:
chicago          0.50
london           0.40
losangeles       0.27
melbourne        0.34
newyork          0.38
sanfrancisco     0.48
singapore        0.48
sydney           0.54
toronto          0.23
dtype: float64
Test Random by category STD:
chicago          0.0270
london           0.0271
losangeles       0.0252
melbourne        0.0241
newyork          0.0134
sanfrancisco     0.0276
singapore        0.0240
sydney           0.0279
toronto          0.0215
dtype: float64
Test Selected:
Accuracy    0.68
dtype: float64
Test Selected by category:
              Accuracy
Categ
chicago           0.53
london            0.56
sydney            0.70
losangeles        0.70
melbourne         0.73
newyork           0.75
toronto           0.79
sanfrancisco      0.80
singapore         0.87
```

Experiment 2:

```
Test Random:
0.41 (+/-0.0241)
Test Random by category MEAN:
chicago          0.56
london           0.39
losangeles       0.30
melbourne        0.34
newyork          0.39
sanfrancisco     0.44
singapore        0.52
sydney           0.52
toronto          0.25
dtype: float64
Test Random by category STD:
chicago          0.0275
london           0.0283
losangeles       0.0222
melbourne        0.0241
newyork          0.0131
sanfrancisco     0.0263
singapore        0.0277
sydney           0.0265
toronto          0.0210
dtype: float64
Test Selected:
Accuracy    0.68
dtype: float64
Test Selected by category:
              Accuracy
Categ
london            0.54
chicago           0.57
losangeles        0.67
newyork           0.69
sydney            0.70
melbourne         0.75
toronto           0.78
sanfrancisco      0.81
singapore         0.88
```

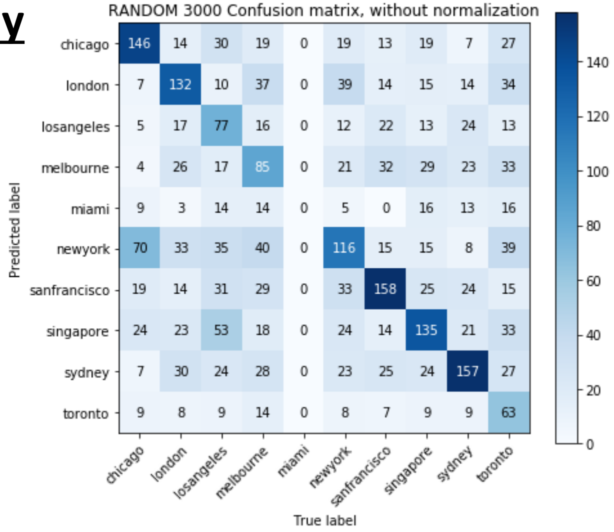
Experiment 3:

```
Test Random:
0.4 (+/-0.0236)
Test Random by category MEAN:
chicago          0.47
london           0.39
losangeles       0.25
melbourne        0.29
newyork          0.41
sanfrancisco     0.46
singapore        0.47
sydney           0.58
toronto          0.26
dtype: float64
Test Random by category STD:
chicago          0.0266
london           0.0259
losangeles       0.0209
melbourne        0.0266
newyork          0.0126
sanfrancisco     0.0284
singapore        0.0239
sydney           0.0251
toronto          0.0226
dtype: float64
Test Selected:
Accuracy    0.67
dtype: float64
Test Selected by category:
              Accuracy
Categ
chicago           0.51
london            0.53
losangeles        0.65
newyork           0.71
sydney            0.71
melbourne         0.74
toronto           0.81
sanfrancisco      0.81
singapore         0.89
```

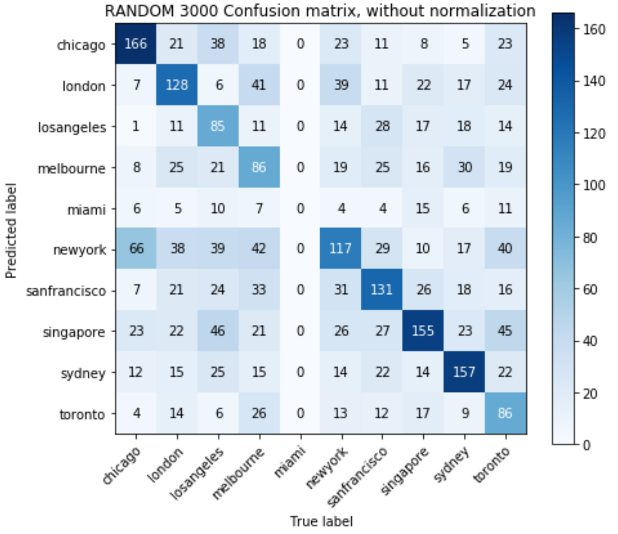
Results – in-depth analysis

10 x Randomly selected 300 from test images:

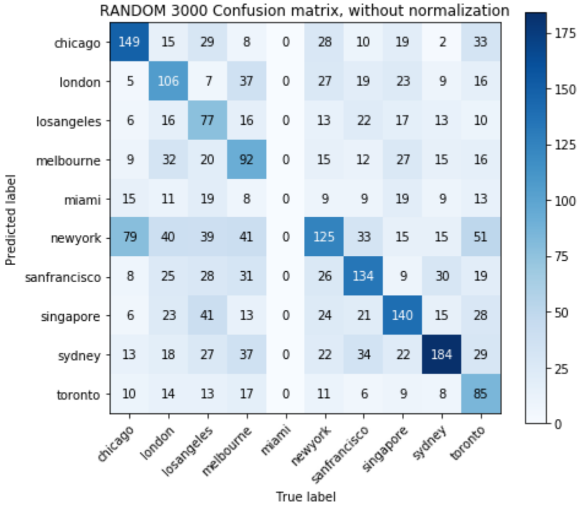
Experiment 1:



Experiment 2:

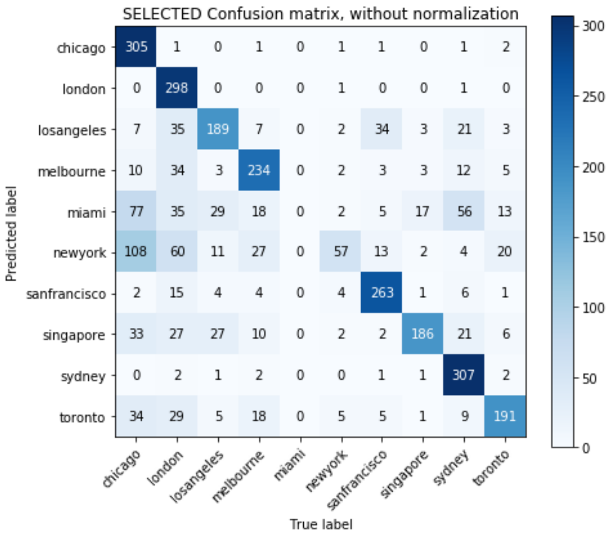


Experiment 3:

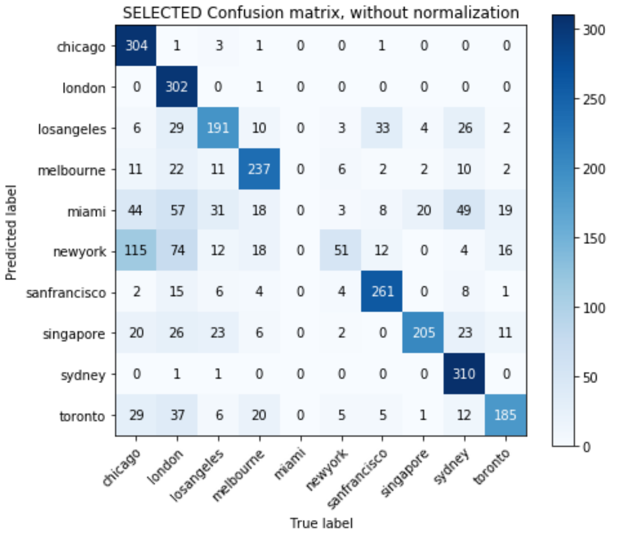


Selected 300 from test images:

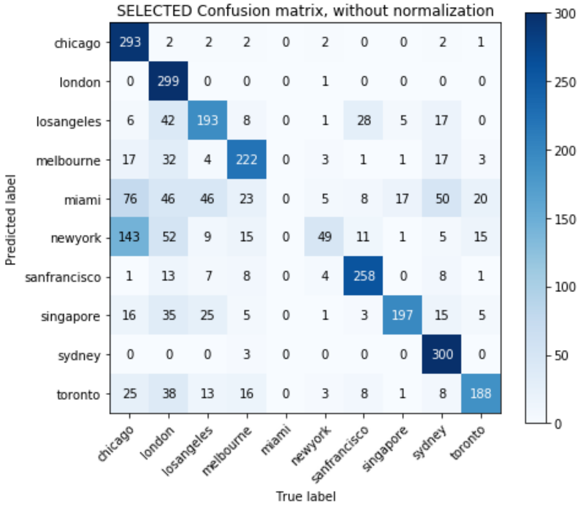
Confusion matrix, without normalization



Confusion matrix, without normalization



Confusion matrix, without normalization



THE END!