Image Recognition Approach for Expediting Chinese Cafeteria Checkout Process

利用影像辨識加快自助餐結帳流程

405850420 吳柏霆
405850206 鄒亞微
405856046 湯智天

指導教授：張峯誠 博士

Outline

Team Works

Work \ Member	吳柏霆	鄒亞微	湯智天
Data Collection	✓	✓	✓
Data Preprocessing			✓
qqwweee / keras-yolo3	✓		
AlexeyAB / darknet		✓	
experiencor / keras-yolo3	✓		
Report	✓	✓	✓
PPT	✓	✓	✓

01

Introduction

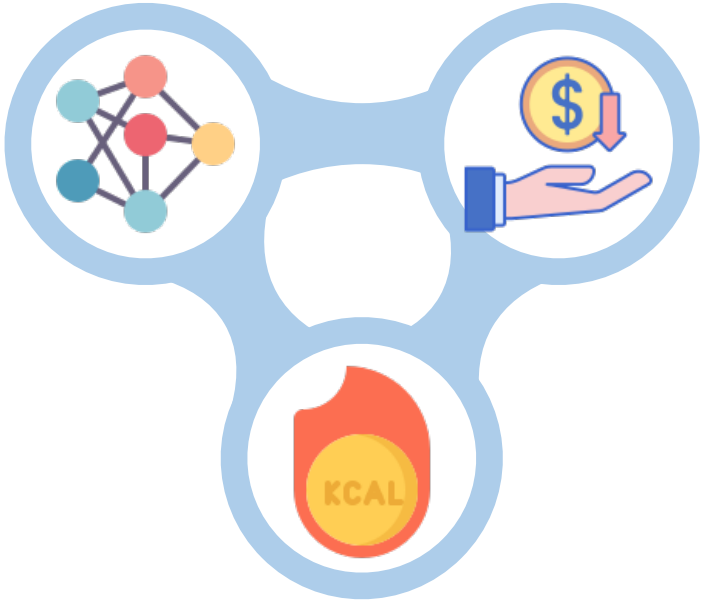
Motivation

A long queue during mealtime in the school cafeteria.

Goal

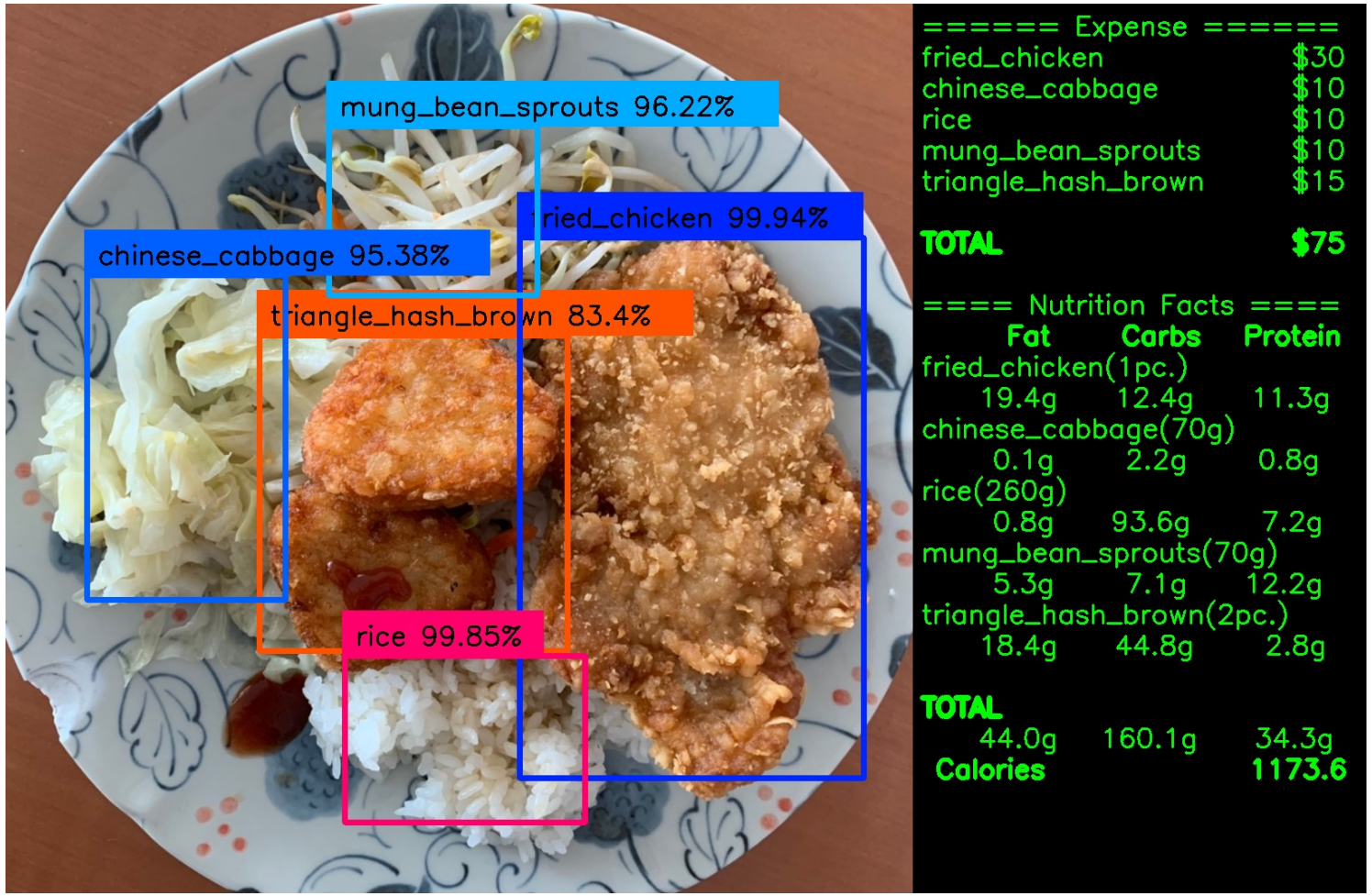
Build up an image recognition system

that automatically calculates price and nutrition facts.



Contribution

We eventually build up the image recognition system.



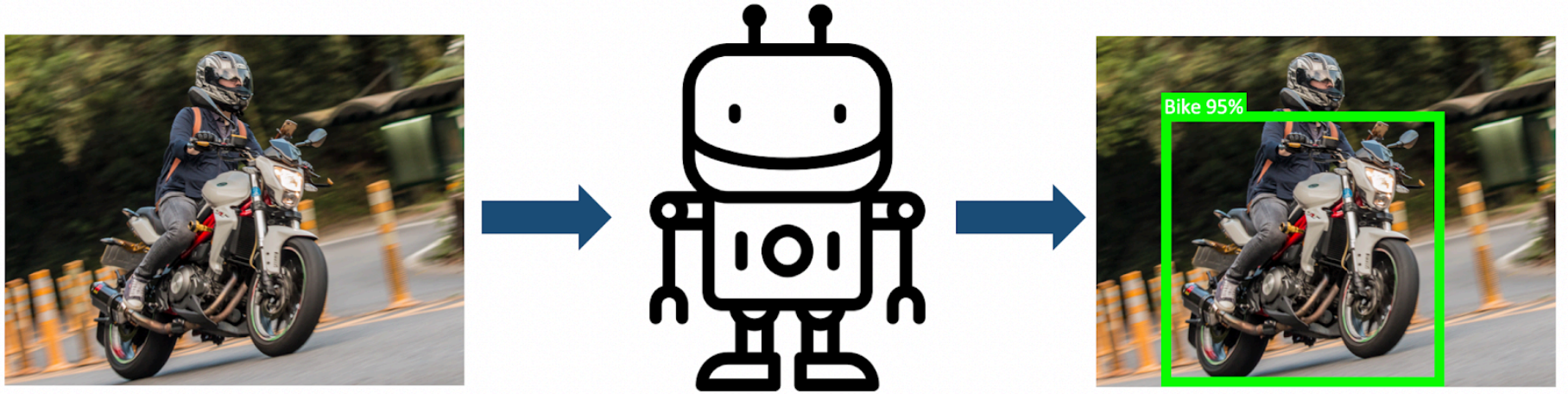
02

YOLOv3

Important concepts of YOLOv3

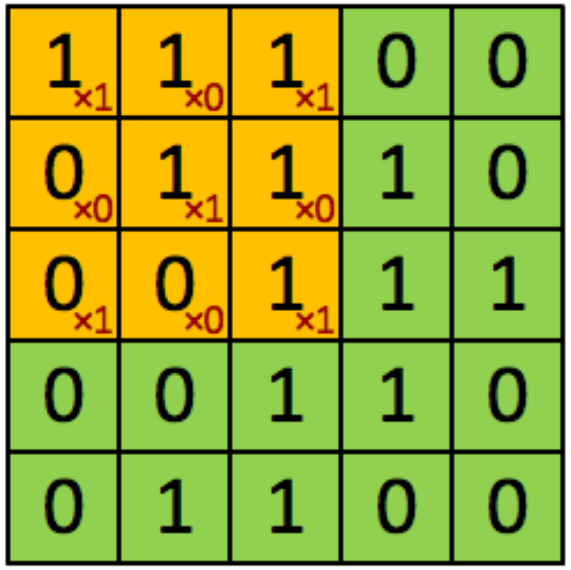
1. Object Detection
2. Convolutional Layer and Filters
3. Residual Block

Object Detection

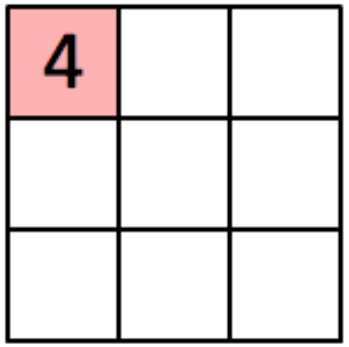


Identifies and locates objects from an image

Convolutional Layer and Filters



Image



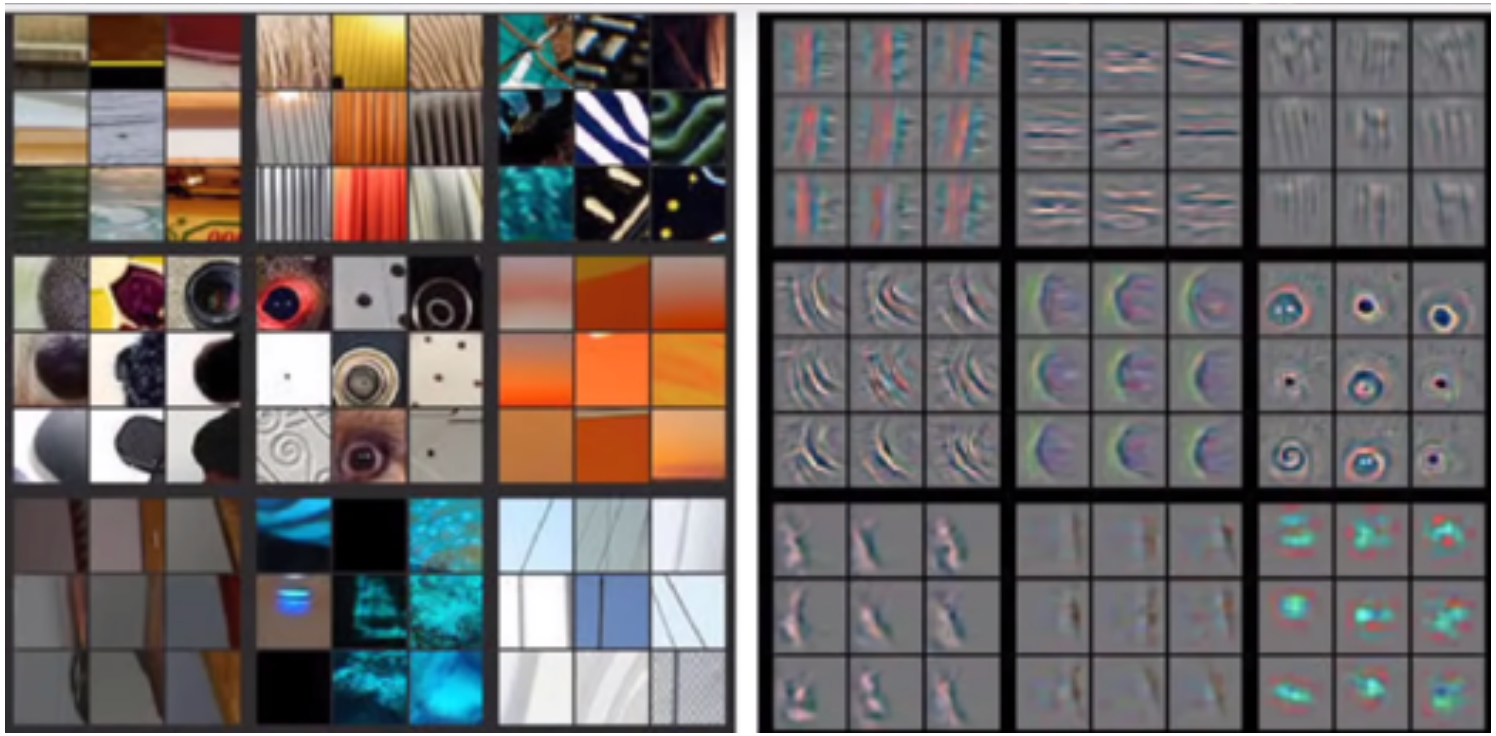
Convolved Feature

Retrieve features from an image

Convolutional Layer and Filters

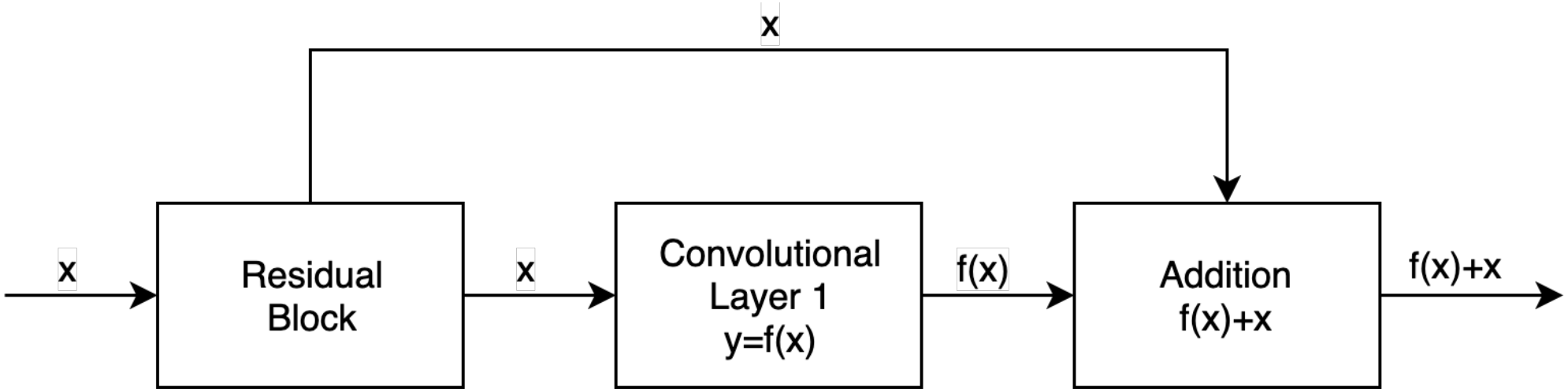
Original images

Convolved features



Different patterns can be retrieved by different filters

Residual Block



A layer will be fed to both the next layer and the layer after the next.

Features can be preserved to the layer after next.

Structure of YOLOv3

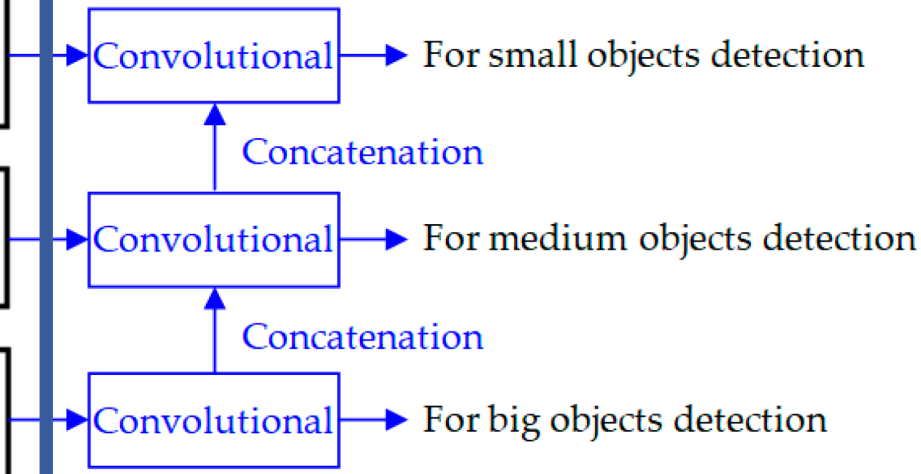
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	416 × 416
	Convolutional	64	3 × 3/2	208 × 208
1 ×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			208 × 208
	Convolutional	128	3 × 3/2	104 × 104
2 ×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			104 × 104
	Convolutional	256	3 × 3/2	52 × 52
8 ×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			52 × 52
	Convolutional	512	3 × 3/2	26 × 26
8 ×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			26 × 26
	Convolutional	1024	3 × 3/2	13 × 13
4 ×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			13 × 13

Base Network

Used to extract features given an input

Detection Network

Predict outputs based on the features extracted from base network



Features of small objects will not lost

Structure of YOLOv3

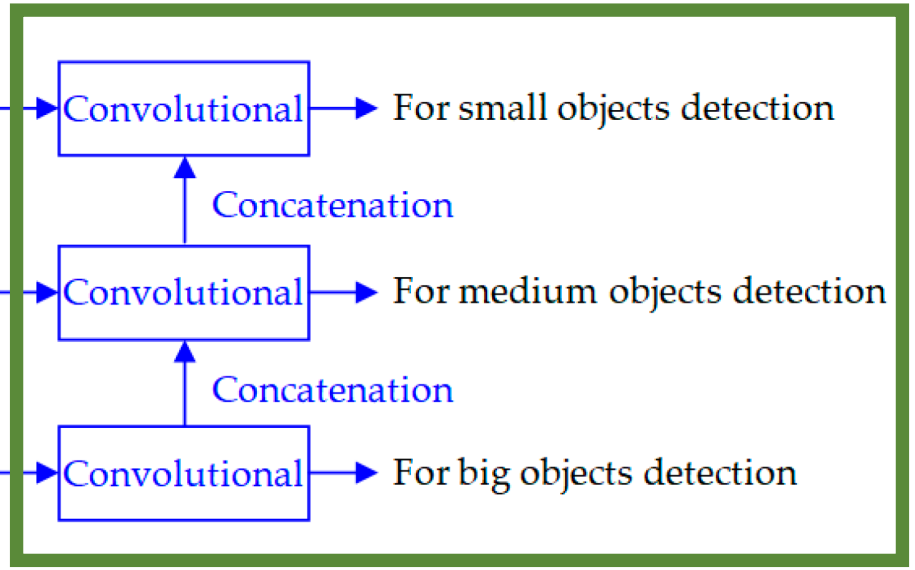
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	416 × 416
	Convolutional	64	3 × 3/2	208 × 208
1 ×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			208 × 208
	Convolutional	128	3 × 3/2	104 × 104
2 ×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			104 × 104
	Convolutional	256	3 × 3/2	52 × 52
8 ×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			52 × 52
	Convolutional	512	3 × 3/2	26 × 26
8 ×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			26 × 26
	Convolutional	1024	3 × 3/2	13 × 13
4 ×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			13 × 13

Base Network

Used to extract features given an input

Detection Network

Predict outputs based on the features extracted from base network



Features of small objects will not lost

Structure of YOLOv3

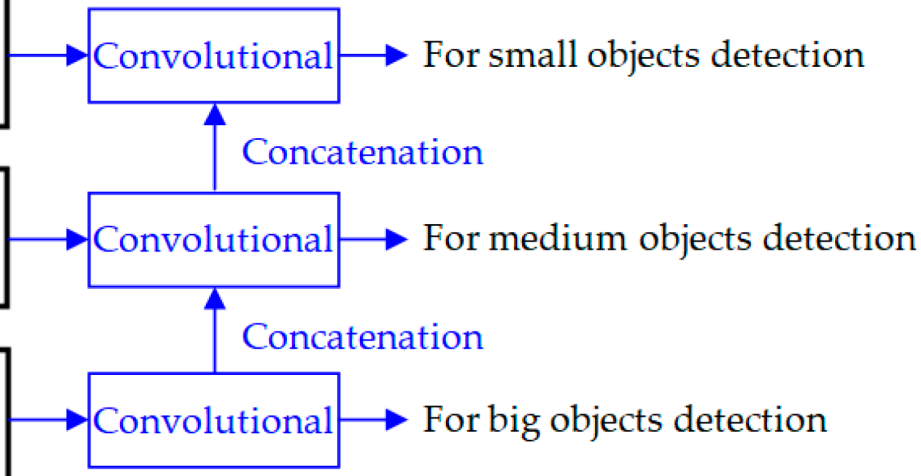
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	416 × 416
	Convolutional	64	3 × 3/2	208 × 208
1 ×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			208 × 208
	Convolutional	128	3 × 3/2	104 × 104
2 ×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			104 × 104
	Convolutional	256	3 × 3/2	52 × 52
8 ×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			52 × 52
	Convolutional	512	3 × 3/2	26 × 26
8 ×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			26 × 26
	Convolutional	1024	3 × 3/2	13 × 13
4 ×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			13 × 13

Base Network

Used to extract features given an input

Detection Network

Predict outputs based on the features extracted from base network



Features of small objects will not lost


03
Data

Data Preparation

1. Data Collection
2. Data Preprocessing
3. Splitting Data

Data Collection

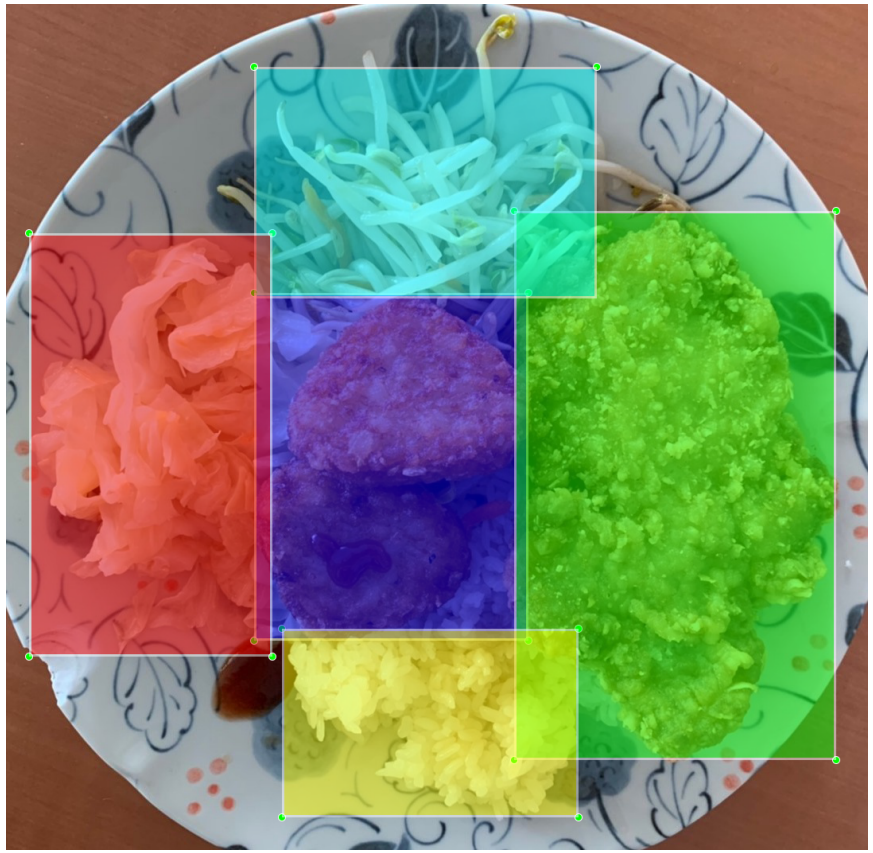
Carrot Eggs
Chicken Nuggets
Chinese Cabbage
Chinese Sausages
Curry
Fried Chicken
Fried Dumplings
Fried Egg
Mung Bean Sprouts
Rice
Triangle Hash Browns
Water Spinach

Limit down to 12 entrees

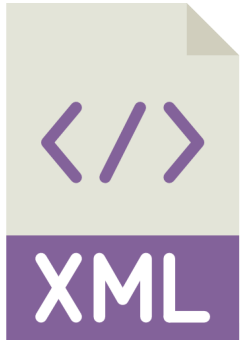
Routinely take sample pictures of those entrees

Data Preprocessing

LabelImg: label out all the entrees



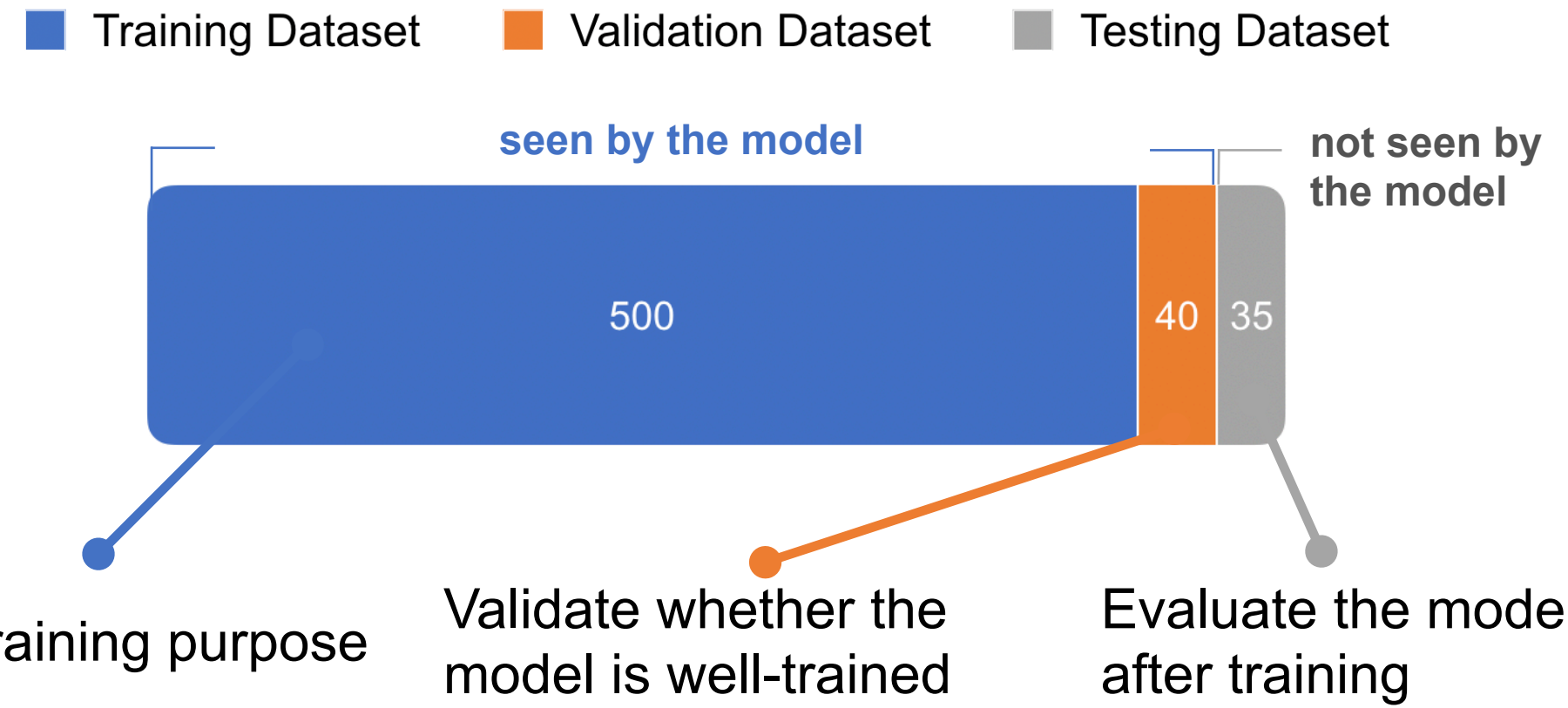
Generates XML files to store the location of entrees




```
...
<object>
    <name>triangle_hash_brown</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    <bndbox>
        <xmin>507</xmin>
        <ymin>658</ymin>
        <xmax>827</xmax>
        <ymax>898</ymax>
    </bndbox>
</object>
...
```

Splitting Data

575 Images Taken in Total



04

Training & Result

Training

Split training dataset to 5 piles

- 100 images per pile
- Incrementally adding the piles to training

100 100 100 100 100

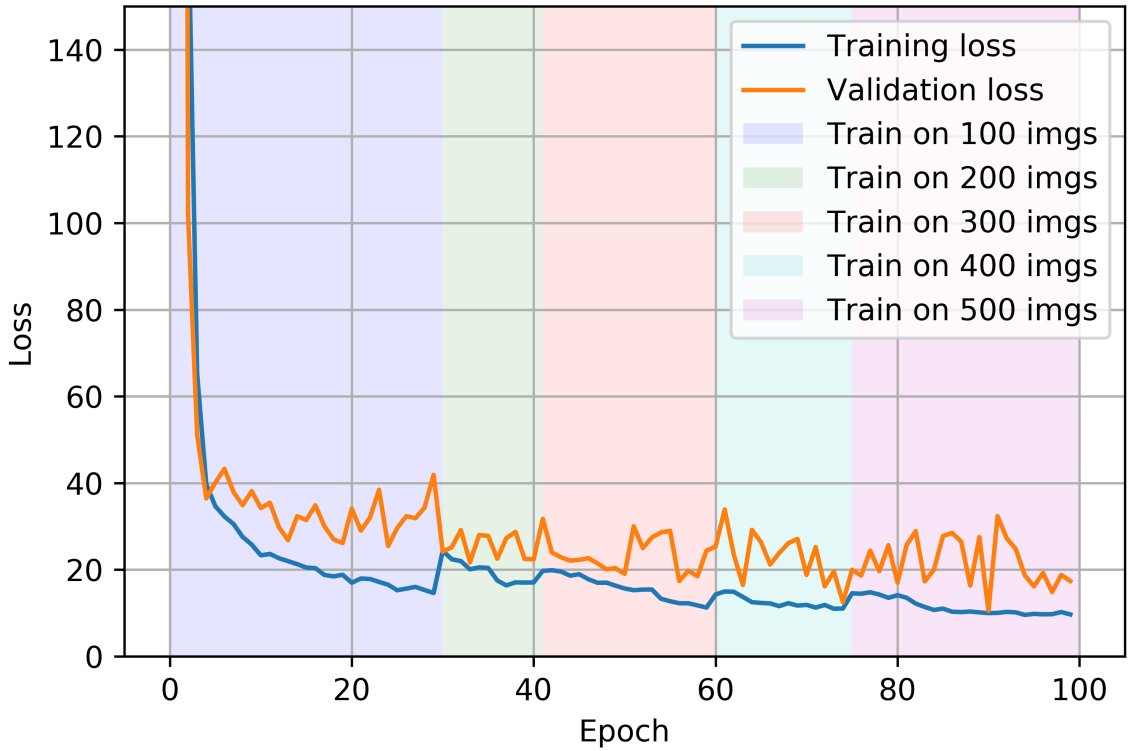
Train on Cloud - Google Colab

Platform for Developing Machine Learning Models

Free GPU (Nvidia K80, K100, P100)

Training Result

Training & Validation Losses



mAP in Testing

```
carrot_eggs: 0.8200
chicken_nuggets: 0.2000
chinese_cabbage: 0.2857
chinese_sausage: 0.7500
curry: 0.8000
fried_chicken: 0.9028
fried_dumplings: 0.6964
fried_eggs: 0.9048
mung_bean_sprouts: 0.7222
rice: 0.7351
triangle_hash_brown: 0.5510
water_spinach: 1.0000
mAP: 0.6973
```

Testing the Correctness of Detecting the 12 Entrees in a Plate

Criteria of detecting

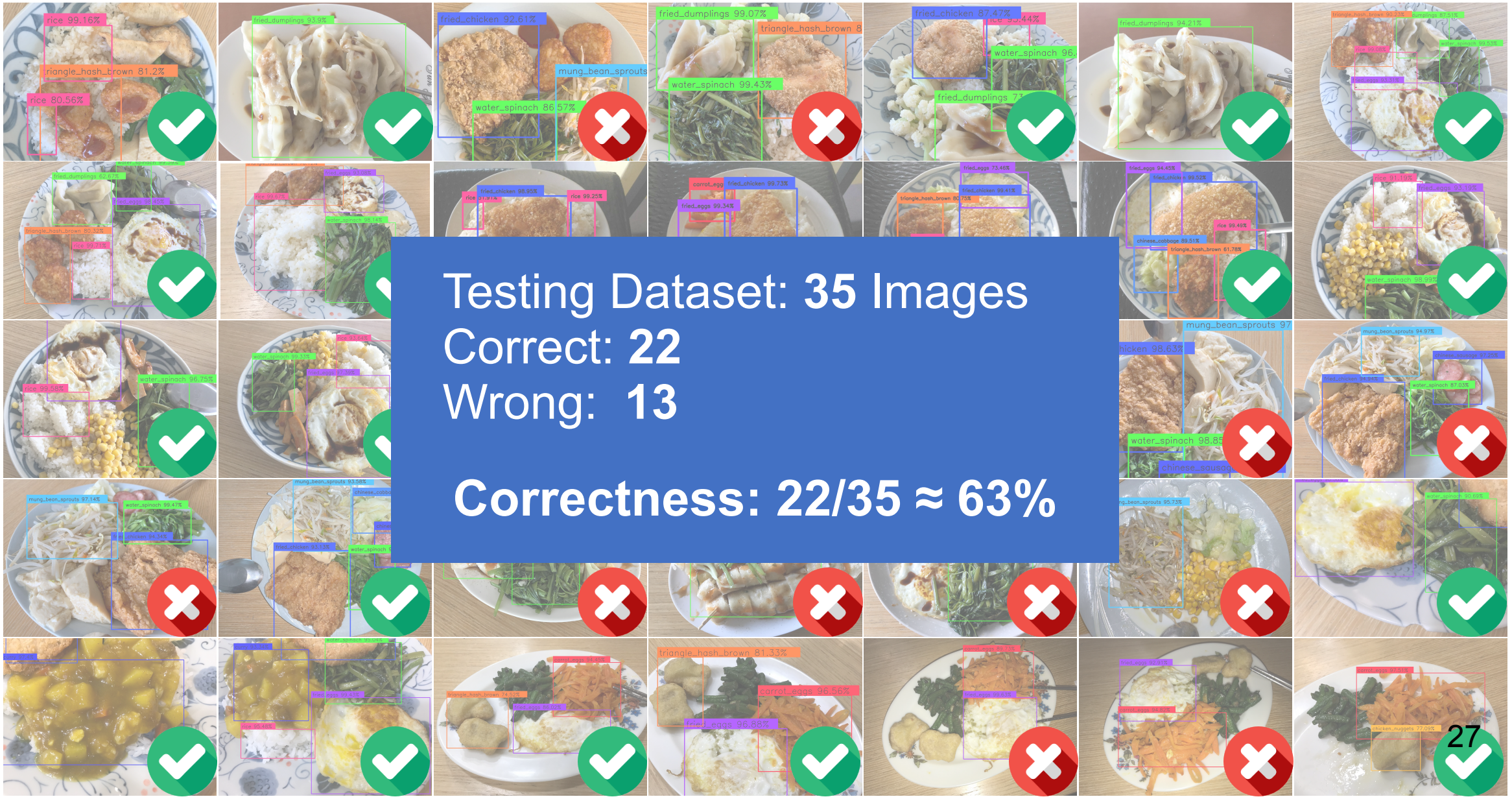
All entrees shown in image are correctly detected.

An entree is shown in the image but doesn't being detected.
(**True Negative**)

Detecting an entree which actually doesn't exist in the image.
(**False Positive**)

Testing the Correctness of Detecting the 12 Entrees in a Plate

Testing the Correctness of Detecting the 12 Entrees in a Plate



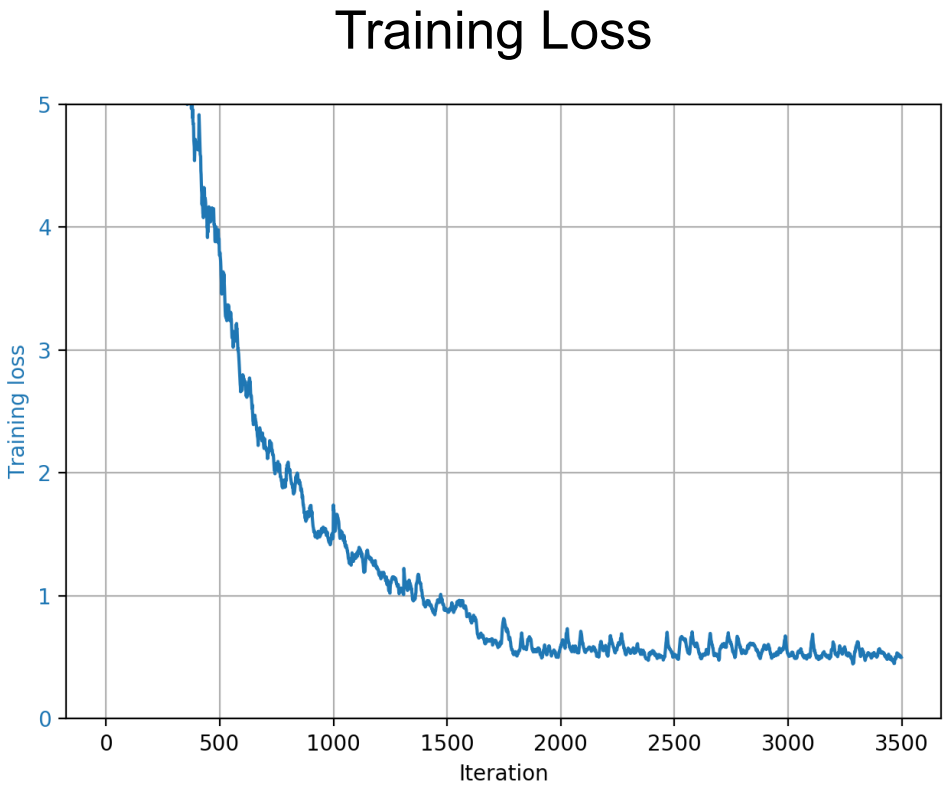
Testing Dataset: **35** Images
Correct: **22**
Wrong: **13**

Correctness: 22/35 ≈ 63%

Additional Tries

AlexeyAB / darknet

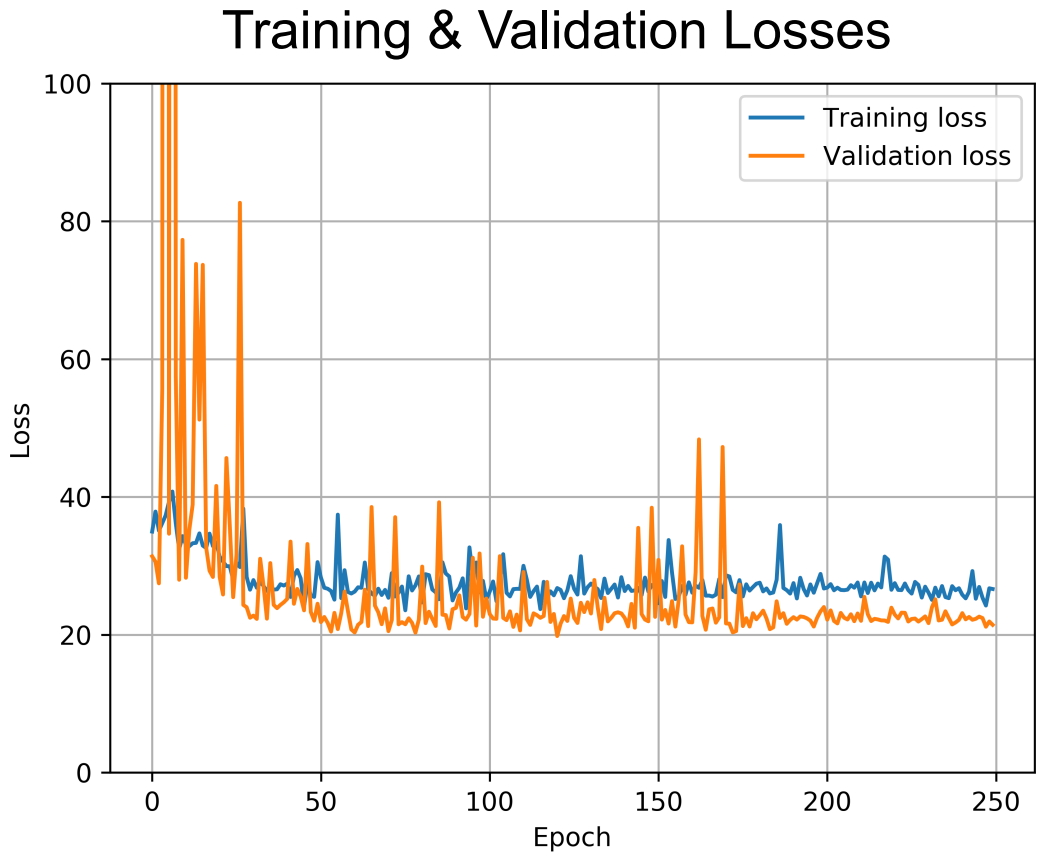
- Training takes too long
- Written in C, hard to proceed further part



Additional Tries

qqwweee / keras-yolo3

- Learning is saturated but unable to predict



Test with testing dataset

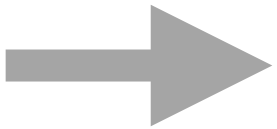
05

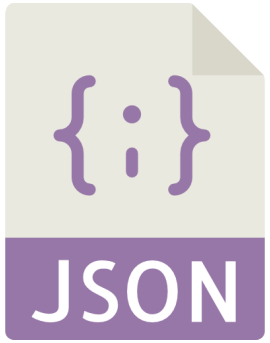
Price & Nutrition Facts

Price

Entree	Price
Carrot Eggs	10
Chicken Nuggets	15
Chinese Cabbage	10
Chinese Sausages	10
Curry	10
Fried Chicken	30
Fried Dumplings	15
Fried Egg	10
Mung Bean Sprouts	10
Rice	10
Triangle Hash Browns	15
Water Spinach	10

Survey the price of 12 entrees

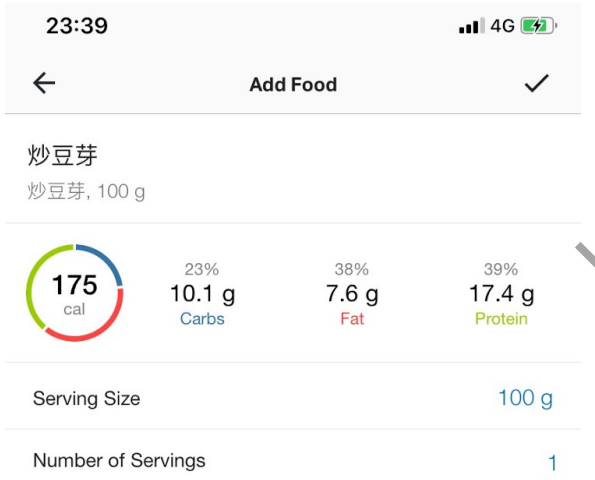




Nutrition Facts

MyFitnessPal

Collect the nutrition facts of 12 entrees



Cafeteria

Measure the weight per serving

Conversion

JSON

Price & Nutrition Facts Table

Entree	Amount per serving	Fat(g)	Carbohydrate(g)	Protein(g)	Price
Carrot Eggs	70g	1.9	4.2	3.2	10
Chicken Nuggets	4pc.	12.0	12.0	9.0	15
Chinese Cabbage	70g	0.1	2.2	0.8	10
Chinese Sausages	24g (6pc.)	7.2	3.8	3.8	10
Curry	170g	7.1	16.5	10.7	10
Fried Chicken	1pc.	19.4	12.4	11.3	30
Fried Dumplings	4pc.	16.0	20.0	8.0	15
Fried Egg	1pc.	15.0	1.4	13.5	10
Mung Bean Sprouts	70g	5.3	7.1	12.2	10
Rice	260g	0.8	93.6	7.2	10
Triangle Hash Browns	4pc.	18.4	44.8	2.8	15
Water Spinach	70g	2.3	2.9	2.6	10

The JSON Config File


```
"entrees": {
    "carrot_eggs":          {"serving_amount":  70, "amount_unit":   "g", "fat":   1.9, "carbohydrate":  4.2, "protein":  3.2, "price": 10},
    "chicken_nuggets":      {"serving_amount":   4, "amount_unit": "pc.", "fat":  12.0, "carbohydrate": 12.0, "protein":  9.0, "price": 15},
    "chinese_cabbage":      {"serving_amount":  70, "amount_unit":   "g", "fat":   0.1, "carbohydrate":  2.2, "protein":  0.8, "price": 10},
    "chinese_sausage":      {"serving_amount":   6, "amount_unit": "pc.", "fat":   7.2, "carbohydrate":  3.8, "protein":  3.8, "price": 10},
    "curry":                {"serving_amount": 170, "amount_unit":   "g", "fat":   7.1, "carbohydrate": 16.5, "protein": 10.7, "price": 10},
    "fried_chicken":        {"serving_amount":   1, "amount_unit": "pc.", "fat":  19.4, "carbohydrate": 12.4, "protein": 11.3, "price": 30},
    "fried_dumplings":      {"serving_amount":   4, "amount_unit": "pc.", "fat":  16.0, "carbohydrate": 20.0, "protein":  8.0, "price": 15},
    "fried_eggs":           {"serving_amount":   1, "amount_unit": "pc.", "fat":  15.0, "carbohydrate":  1.4, "protein": 13.5, "price": 10},
    "mung_bean_sprouts":    {"serving_amount":  70, "amount_unit":   "g", "fat":   5.3, "carbohydrate":  7.1, "protein": 12.2, "price": 10},
    "rice":                 {"serving_amount": 260, "amount_unit":   "g", "fat":   0.8, "carbohydrate": 93.6, "protein":  7.2, "price": 10},
    "triangle_hash_brown":  {"serving_amount":   4, "amount_unit": "pc.", "fat":  18.4, "carbohydrate": 44.8, "protein":  2.8, "price": 15},
    "water_spinach":        {"serving_amount":  70, "amount_unit":   "g", "fat":   2.3, "carbohydrate":  2.9, "protein":  2.6, "price": 10}
}
```

06

Demonstration

07

Conclusion

Conclusion

An extraordinary motivation with an innovative solution being invented.

- Compare and Select Model Structure
- Collect and Prepare Data
- Training with Different Framework
- Collect Price, Amount per Serving and Nutrition Facts Info.
- Combine Everything Together -

  A Realtime Automatic Price and Nutrition Facts Calculation System

Expedite the Queue can be a Realistic Possibility!!

Thanks for Listening

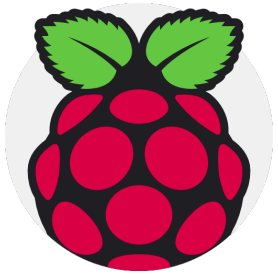
405850420 吳柏霆
405850206 鄒亞微
405856046 湯智天

指導教授：張峯誠 博士

Conclusion

An extraordinary motivation with an innovative solution being invented.

Coral



But some things yet to be improved:

- Correctness should be enhanced
- Transplant to Raspberry Pi
- USB accelerator (Google Coral) for boosting the inference speed

Expedite the Queue can be a Realistic Possibility!!