ngee ann
school of infocomm technology

Deep Learning DLIR Assignment

Specialist Diploma in applied Generative AI (SDGAI)
Apr 2025 Semester

Presented by: Lim Ai Sim Elizabeth (3440680F)

Presented on 09/06/2025

Assignment Objectives

Main objective: To design and implement a convolutional neural network (CNN) based model that can accurately classify food images into one of the 10 specified food categories.

Approaches:

1. Customized CNN model (“Build from scratch”)
2. Use a pretrained model (Apply transfer learning)
3. Use the best model on some images taken from the internet.

Environment setup

Work in cloud computing platforms:

- Google Colab
- AWS Sagemaker AI

Run training and analyse in Jupyter Notebook

- TensorFlow / Keras libraries
- matplotlib

Data exploration

Step 1: Unzip the image data into a folder, data processing to use the data for training, validation and testing.

Step 2: Check the contents and examine images

- Check the number of images provided for training and testing. (750 images per class for training, 250 images per class for validation, 50 images per class set aside for test evaluation.

Customized CNN Model Design

1) Decide on the initial architecture

Sequential model

- 3-4 convolutional + max pooling layers for feature extraction.
- 2-3 fully connected dense layers for classification task.

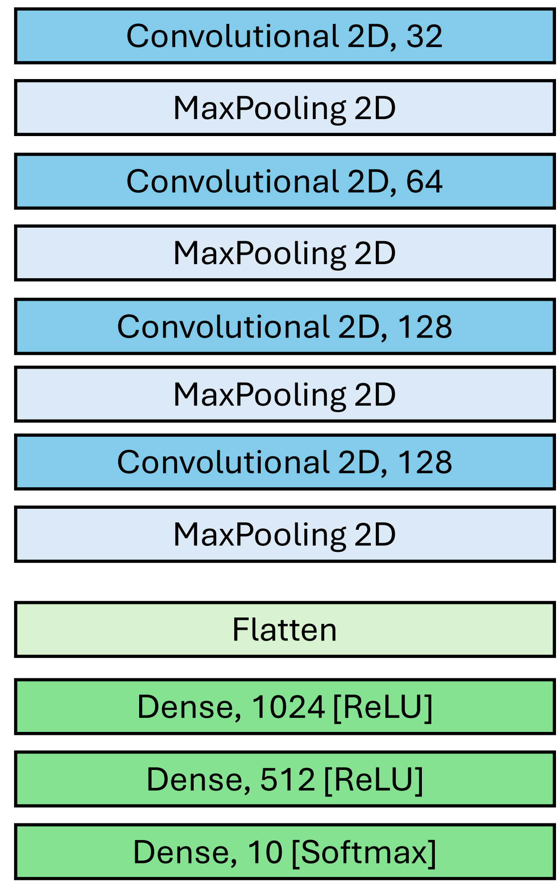
2) Decide on training and tweaking strategy

- Train the model and monitor progress using accuracy and loss graphs for training and validation.
- Train without augmenting the data. Mitigate overfitting. Perform data augmentation and use that to further improve the model.
- Change one component at a time to observe the effects of the change.

Customized CNN Model Training

Training format:

- Batch size = 25
- Training epochs = 30-80
- Optimizer: RMSprop
- Loss: Categorical cross-entropy
- Metric: Accuracy
- Learning rate: 0.0001 (1e-4)



Customized Model Design and Adjustments (I)

Strategy: Adjust and optimize without data augmentation.

Without data augmentation

Model 1-1 → Model 1-2 → Model 1-3 → Model 1-4 → **Model 1-5** → Model 1-6 → Model 1-7

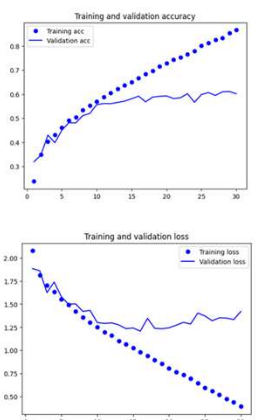
Model 1-2: Add dropout

Model 1-3: Changed kernel size from 3x3 to 5x5

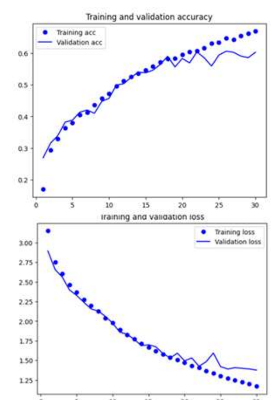
Model 1-4: Add L2 regularization

Model 1-5: Add a final convolutional layer with 256 output channels

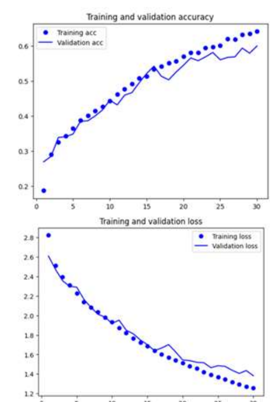
Model 1-7: Reduce one Dense layer (1024 to 512)



Test accuracy = 0.542



Test accuracy = 0.608



Test accuracy = 0.594

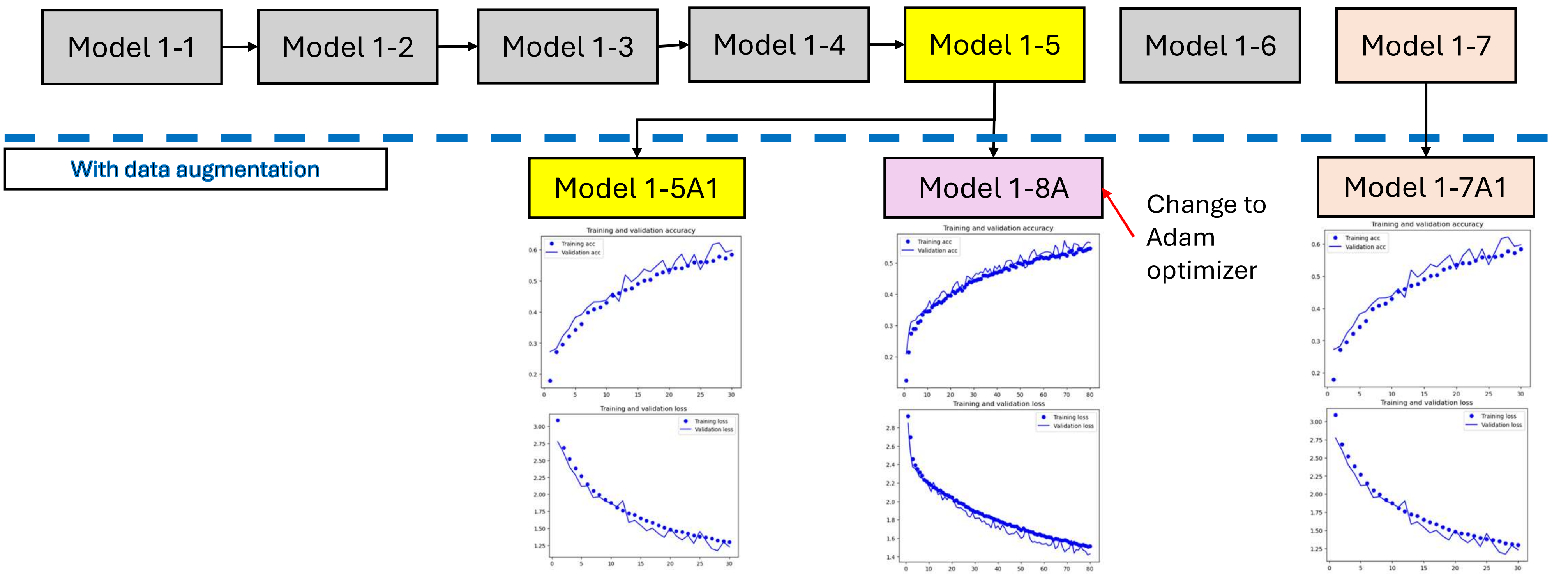
Data augmentation

- Data augmentation increases training data pool by generating variations to existing training data.
- Performed with consideration to realistic variations.

<table>
<tr><th>Code</th><th>Explanation</th></tr>
<tr><td><pre>train_datagen = ImageDataGenerator(
      rescale=1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.3,
      horizontal_flip=True,
      brightness_range=[0.8,1.2],
      fill_mode='nearest')</pre></td><td>Rotation angle = 40 degrees left or right
Shift range is in percentage.0.2 = 20% width or height position shift.
Shear range is in radians.
Zoom range is in percentage (0.3 = 30%)
Brightness is in a range 0.8 -1.2 means 20% darker to 20% brighter than original image.</td></tr>
</table>

Customized Model Design and Adjustments (II)

Strategy: Adjusting best models from previous round. (Subsequent slides show the progression of training and model selection.



Test accuracy = 0.686

Test accuracy = 0.5780

Test accuracy = 0.678

Customized Model Design and Adjustments (III)

Model 1-5

Model 1-7

With data augmentation

Model 1-5A1

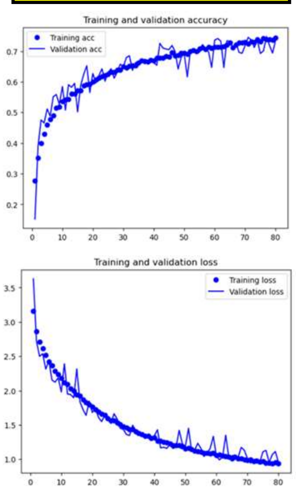
Model 1-8A

Model 1-7A1

Model 1-5A2

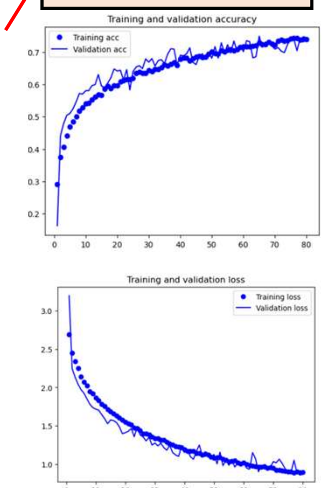
Model 1-7A2

Add batch normalization



Test accuracy = 0.7300

Add batch normalization



Test accuracy = 0.728

Customized Model Design and Adjustments (IV)

Model 1-5

With data augmentation

Model 1-5A1

Add batch normalization

Model 1-5A2

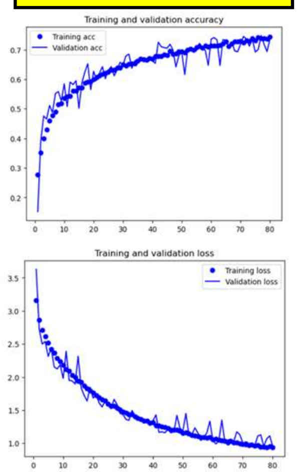
Changed to all ReLU to GeLU activation

Change to Adam optimizer

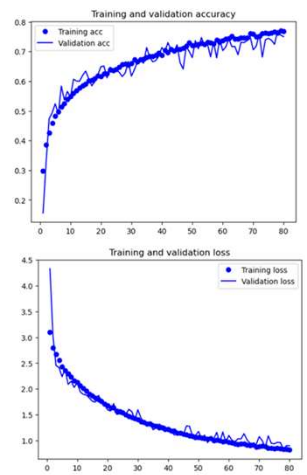
Model 1-5A3

Dense layers changed back to ReLU activation

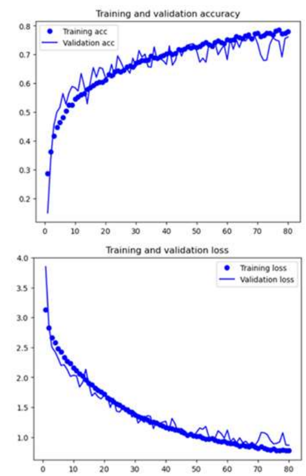
Model 1-5A4



Test accuracy = 0.7300

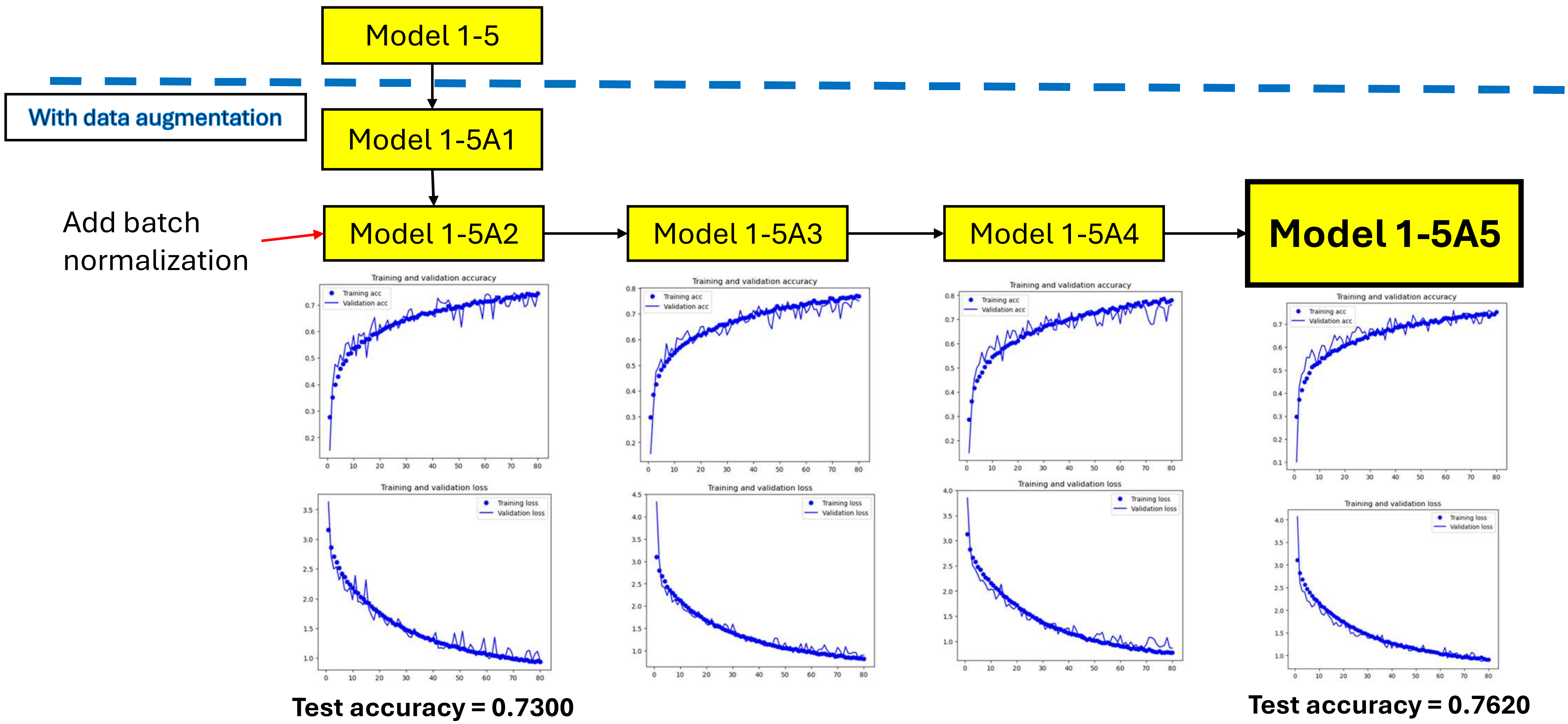


Test accuracy = 0.742



Test accuracy = 0.778

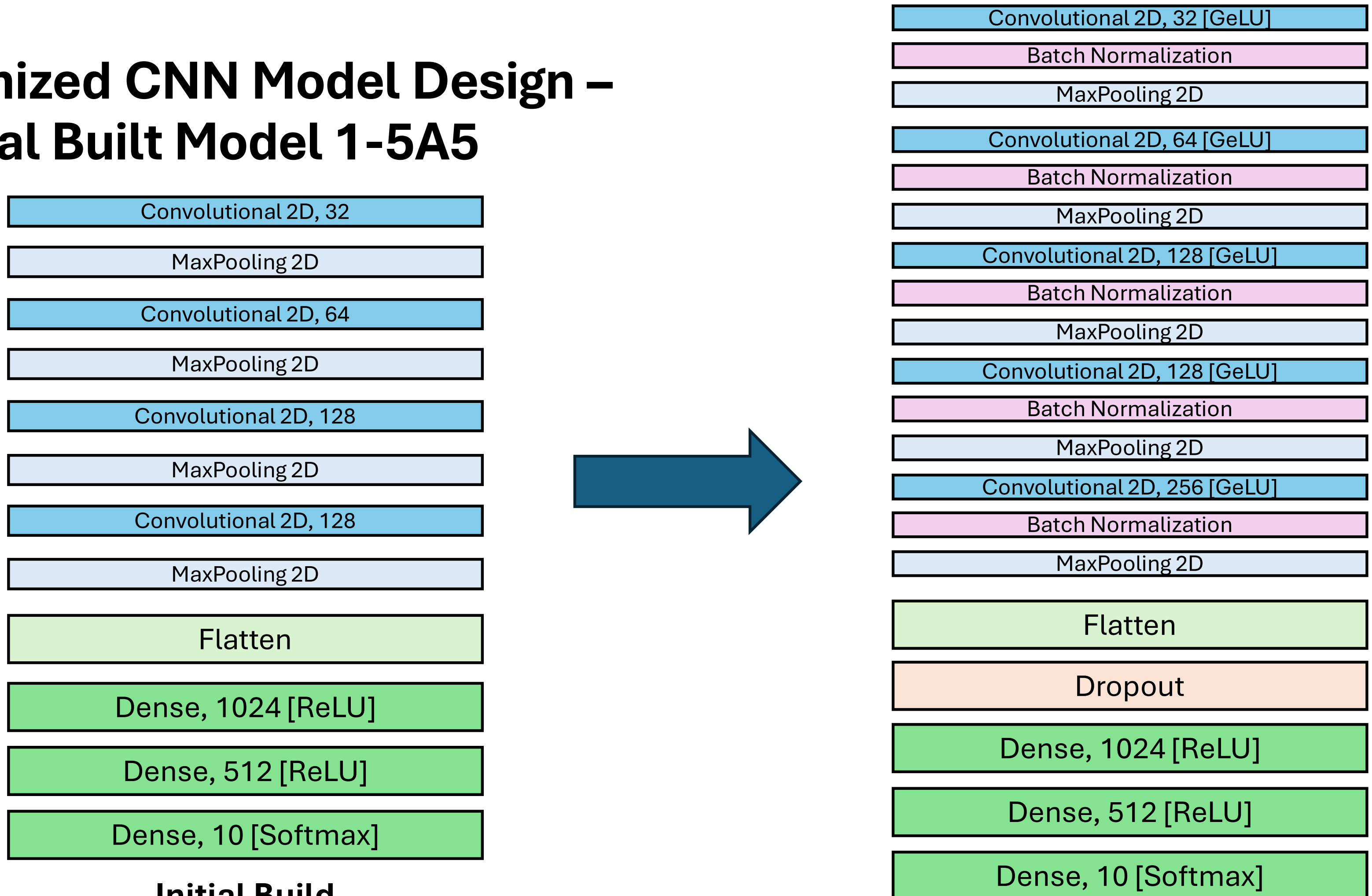
Customized Model Design and Adjustments (V)



Summary of adjustments to Custom CNN Model Design

- Changing number of units in the Dense layers
- Adding a dropout layer and L2 regularization
- Adjusting the kernel size in the convo2D layer from 3x3 to 5x5
- Adding convo2D and MaxPooling2D blocks
- Adding batch normalization layers
- Changing the activation function from ReLU to GeLU
- Adjusting learning rate (1e-4 to 2e-5)
- Using different optimizers (RMSprop vs Adam)

Customized CNN Model Design – The Final Built Model 1-5A5



Initial Build
Test accuracy = 0.542

Final Build: Model 1-5A5
Test accuracy = 0.7620

Transfer Learning with Pretrained Model

- Deciding on which pretrained model to use:

1) Resnet-50
2) InceptionV3

Both models have been used for food classification, but have different internal convolutional structures.

- **Resnet** models have skipped connections to overcome vanishing gradients.
- **Inception models** have convolutional blocks arranged in parallel to capture various features and patterns and to achieve computation efficiency.

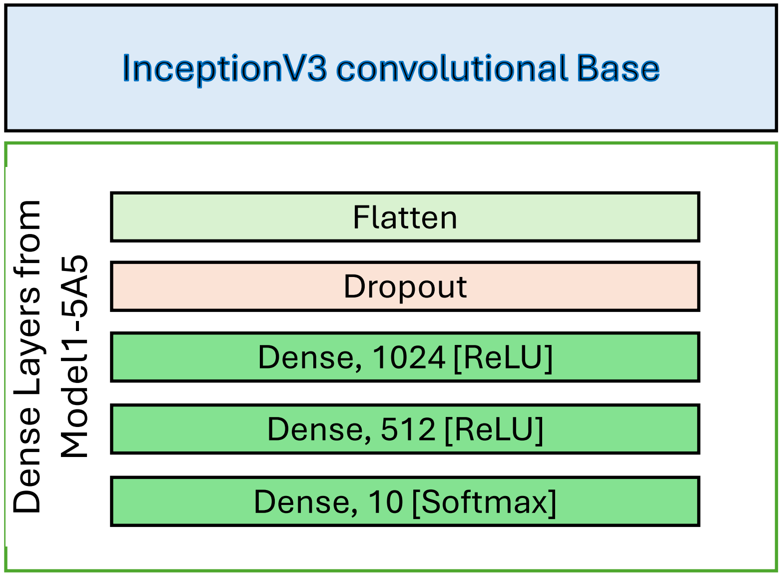
Selected **InceptionV3** as its architecture is very different from own custom CNN model and its convolutional modules suitable for capturing the image details of food.

Transfer Learning with Pretrained Model

Structure: InceptionV3 convolutional base + classification layers from Model 1-5A5.

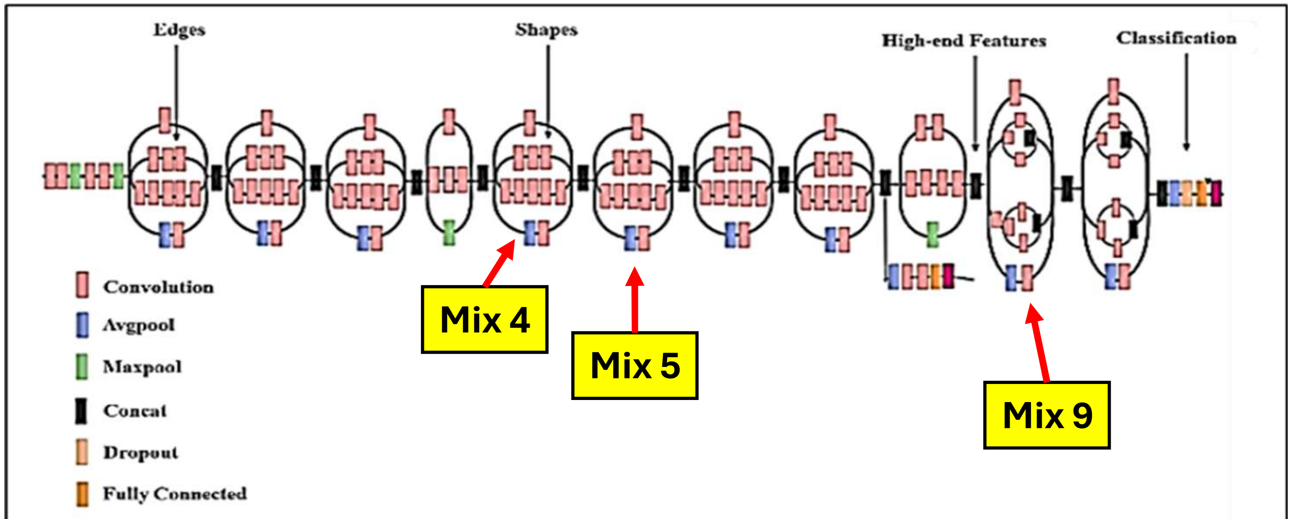
Training and adaptation strategy:

- Integrate pretrained model and custom model layers.
- Train our classifier with augmented training data.
- Finetune the model by unfreezing parts of the convolutional base.



Finetuning – Unfreezing the convolutional modules in InceptionV3

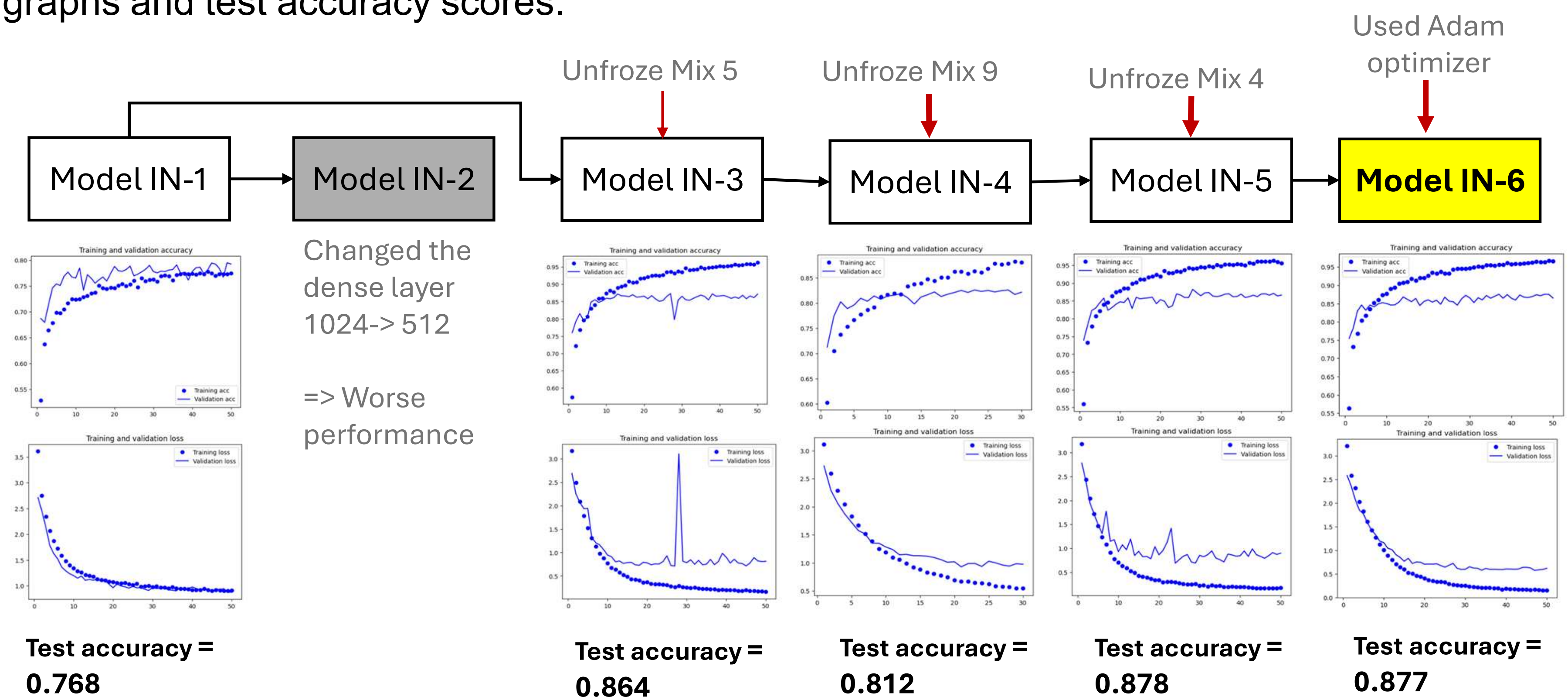
There are **11 inception blocks** (or 'mix' numbered 0 to 10). Each progressive module in the model is learning more and more complex features (Burkapalli *et al* (2020)), from edges to shapes to higher features.



Source: Burkapalli et al (2020)

Transfer Learning Model Design and Adjustments

Strategy: finetuning by unfreezing each Inception module and see its effect on accuracy and loss graphs and test accuracy scores.



Final Test Evaluation

From the final test evaluation accuracy and loss scores, Model ‘**Food_IN6’** is the better model.

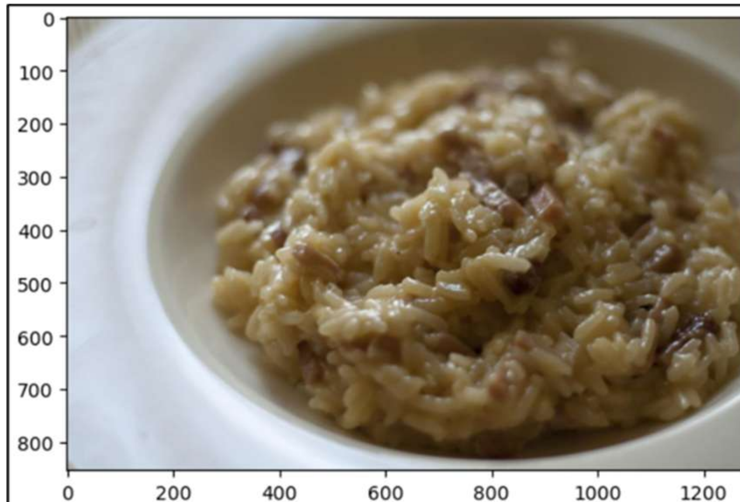
Model	Test accuracy	Test Loss score
1-5A5 [Built-from-scratch CNN model]	0.7659	0.8871
Model Food_IN6 [InceptionV3 convolutional base + customized Dense layers]	0.8750	0.5881

Testing with images from the internet


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 6s 6s/step
The prediction is:  hamburger

     beet_salad  beignets  eggs_benedict  hamburger  hot_and_sour_soup  \
0  8.190054e-07  0.000007       0.000018   0.999964       1.280554e-07

   huevos_rancheros       lasagna       risotto  seaweed_salad  \
0      3.150317e-07  5.229606e-07  1.670149e-07       0.000008

   strawberry_shortcake
0              0.000002
```

Correct!

Hamburger (0.999)

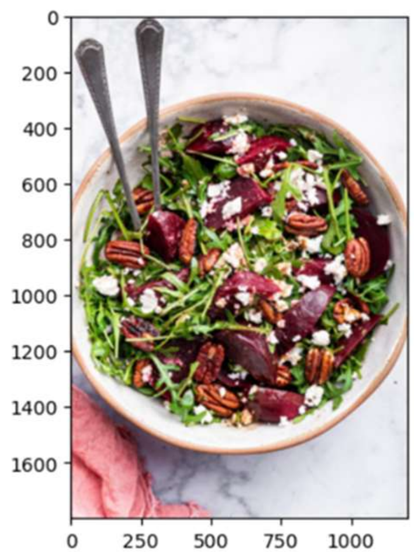



```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 55ms/step
The prediction is:  risotto

     beet_salad      beignets  eggs_benedict     hamburger  hot_and_sour_soup  \
0  2.382025e-10  3.067592e-09   3.660656e-10  1.686473e-09       2.588032e-10

   huevos_rancheros       lasagna  risotto  seaweed_salad  \
0      4.616243e-11  5.100580e-08      1.0   4.861102e-10

   strawberry_shortcake
0          4.884877e-10
```

Correct!

Risotto (1.0)

Testing with images from the internet (II)




```
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 56ms/step
The prediction is:  beet_salad

   beet_salad  beignets  eggs_benedict  hamburger  hot_and_sour_soup  \
0    0.951293  0.000003       0.000005   0.000013            0.02583

   huevos_rancheros   lasagna   risotto  seaweed_salad  strawberry_shortcake
0          0.000008  0.000002  0.000034       0.022802               0.00001
```

Correct!

Beet Salad (0.951)


```
1/1 ━━━━━━━━━━━━━━━━━━━━ 8s 8s/step
The prediction is:  seaweed_salad

     beet_salad      beignets  eggs_benedict     hamburger  hot_and_sour_soup  \
0  8.635825e-07  5.498294e-08   1.211709e-09  2.164986e-08           0.062387

   huevos_rancheros       lasagna   risotto  seaweed_salad  \
0          0.000031  1.941014e-08  0.000016       0.937565

   strawberry_shortcake
0          1.055280e-10
```

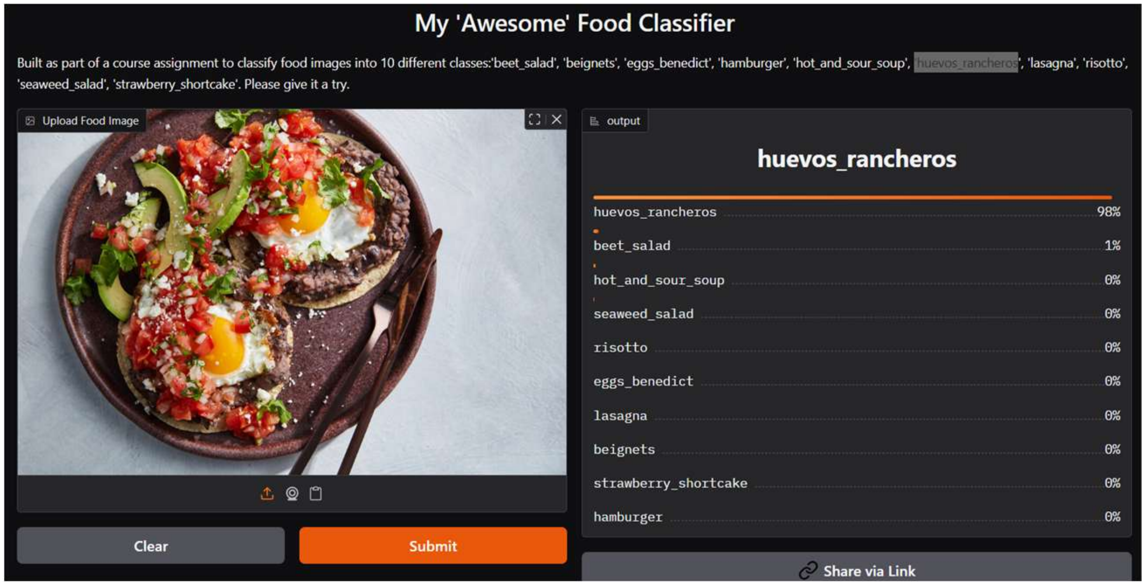
Correct!

Seaweed Salad (0.937)

Model Demo

- The model is available for testing on Hugging Face's Spaces:

https://huggingface.co/spaces/Ravenblack7575/DLIRfood



Limitations and recommendations

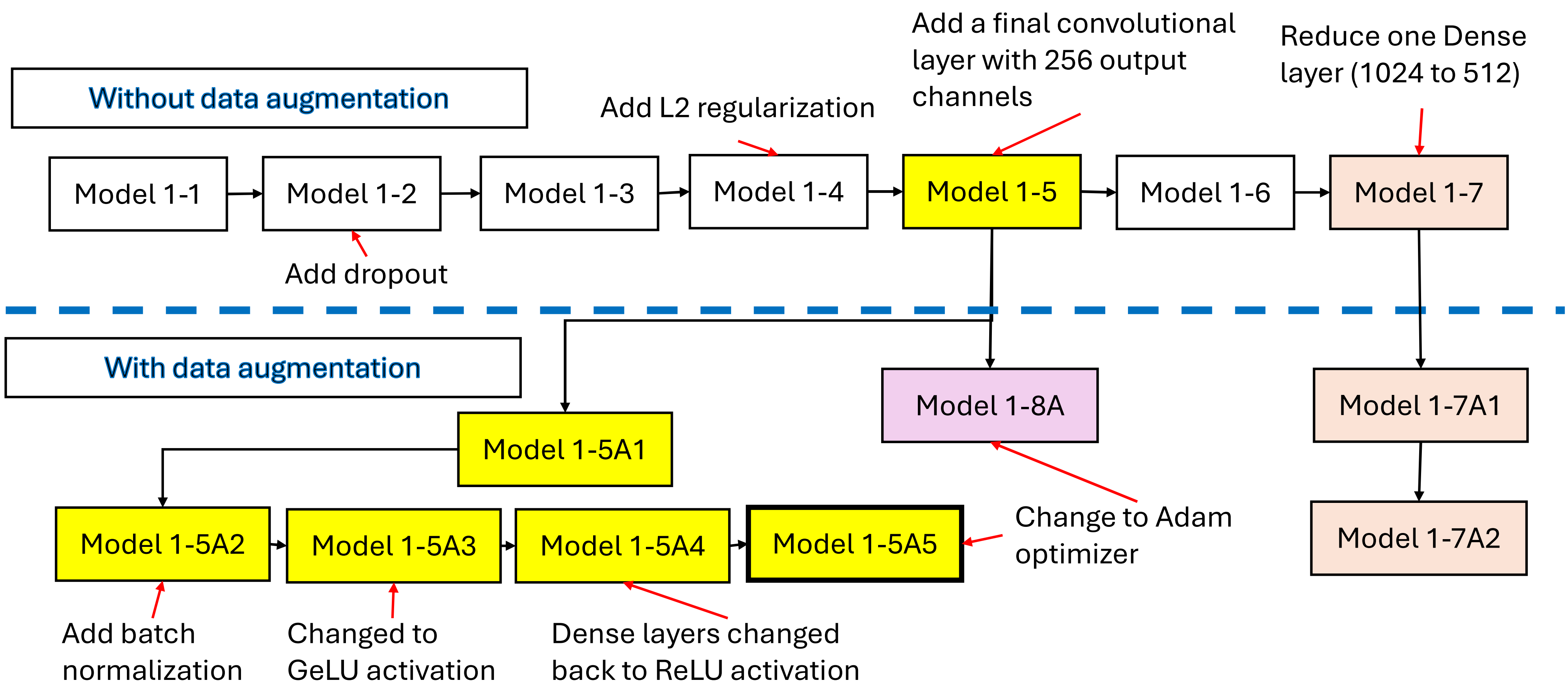
- Currently, if the model was given an image that doesn't fall into any one of these categories, it would still give a prediction based on these ten classes. One way to mitigate this is to implement prediction threshold score where if an image scores are low for this score, then the result could be 'unclassifiable'.

- Functions could be written so that the codes used for model training could be neater and more organized, which may translate to easier execution.

- In future model training, data augmentation will be performed before training commences.

Thank you!

Customized Model Design and Adjustments



Customized Model Design and Adjustments