



# **Model Development Phase Template**

Date	15 March 2024
Team ID	SWTID1720333657
Project Title	Wce Curated Colon Disease Classification Using Deep
Maximum Marks	10 Marks

## Initial Model Training Code, Model Validation and Evaluation Report

The initial model training code will be showcased in the future through a screenshot. The model validation and evaluation report will include a summary and training and validation performance metrics for multiple models, presented through respective screenshots.

## **Initial Model Training Code (5 marks):**

### **Data Augmentation:**





#### VGG16 Model:

```
[1]: import tensorflow as tf
      from tensorflow import keras
      from tensorflow.keras.applications.vgg16 import VGG16
      from tensorflow.keras.layers import Flatten
      from tensorflow.keras.models import Model
      from tensorflow.keras.layers import Dense
      from tensorflow.keras.activations import softmax
      from keras.api import activations
 [2]: Image_size = [224,224]
      sol = VGG16(input_shape = Image_size + [3] ,include_top = False)
      for i in sol.layers:
          i.trainable = False
      y = Flatten()(sol.output)
[3]: final = Dense(4, activation = 'softmax')(y)
                                                                                                                                    □ ↑ ↓ 古 〒 🗎
     vgg16_model = Model(inputs=sol.input, outputs=final)
11]: vgg16_model.summary()
# Assuming L2 weight of 0.01
loss_weight = 0.05
vgg16_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'], loss_weights=loss_weight)
from tensorflow.keras.callbacks import EarlyStopping
                                                                                                                                    ◎ ↑ ↓ 占 무 🛢
 # Set up early stopping to monitor validation accuracy
 early_stopping = EarlyStopping(monitor='val_accuracy', patience=3, mode = 'max')
vgg16_model.fit(train_data, epochs = 15, validation_data = test_data, callbacks = [early_stopping])
[38]: vgg16_model.save('cnn.h5')
      WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We
      recommend using instead the native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.
[52]: from keras.preprocessing import image
      from keras.applications.vgg16 import preprocess_input
      from tensorflow.keras.preprocessing.image import load_img, img_to_array
      import numpy as np
[53]: labels = ['0_normal','1_ulcerative_colitis','2_polyps','3_esophagitis']
      img_path = 'C:/Users/DNIN/Desktop/machine learning tutorial/test/1_ulcerative_colitis/test_ulcer_ (164).jpg'
      #"C:\Users\DNIN\Desktop\machine learning tutorial\test\1_ulcerative_colitis\test_ulcer_ (164).jpg"
      img = load_img(img_path, target_size=(224,224))
      x = img_to_array(img)
      x = preprocess_input(x)
      preds = vgg16_model.predict(np.array([x]))
```





#### **ResNet-50 Model:**

```
[4]: import tensorflow as tf
     from tensorflow import keras
     from tensorflow.keras.applications.resnet50 import ResNet50
     from tensorflow.keras.layers import Flatten
     from tensorflow.keras.models import Model
[6]: Image_size = [224, 224]
     # Load the pre-trained ResNet-50 model
     resnet50 = ResNet50(input_shape=Image_size + [3], include_top=False)
     # Freeze all layers to prevent training
     for layer in resnet50.layers:
        layer.trainable = False
     # Flatten the output from ResNet-50
    y = tf.keras.layers.Flatten()(resnet50.output)
     Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/resnet/resnet50_weights_tf_dim_ordering_tf_kernels_notop.h5
     94765736/94765736 -
[16]: # Assuming L2 weight of 0.01
       loss_weight = 0.01
       resnet50 model.compile(optimizer='adam', loss='categorical crossentropy', metrics=['accuracy'], loss weights=loss weight)
[21]: from tensorflow.keras.callbacks import EarlyStopping
       # Set up early stopping to monitor validation accuracy
       early_stopping = EarlyStopping(monitor='val_accuracy', patience=3)
       resnet50_model.fit(train_data, epochs=20, validation_data=test_data, callbacks=[early_stopping])
```

## **EfficientNet Model:**

```
[7]: from tensorflow.keras.applications import EfficientNetB0
     import tensorflow as tf
     from tensorflow import keras
      from tensorflow.keras.layers import Flatten
      from tensorflow.keras.models import Model
     Image_size = [224, 224]
     efficientnet_model = EfficientNetB0(weights='imagenet', include_top=False, input_shape=Image_size + [3])
     for layer in efficientnet_model.layers:
       layer.trainable = False
      Downloading data from https://storage.googleapis.com/keras-applications/efficientnetb0_notop.h5
     16705208/16705208
                                              - 4s Ous/step
  [7]: from tensorflow.keras.layers import Dense
        \textbf{from} \ \texttt{tensorflow.keras.activations} \ \textbf{import} \ \texttt{softmax}
        \textbf{from} \text{ keras.api } \textbf{import} \text{ activations}
        y = tf.keras.layers.Flatten()(efficientnet_model.output)
        num_classes = 4 # Adjust based on your dataset
        final layer = Dense(num classes, activation='softmax')(y)
         efficientnet_model = Model(inputs=efficientnet_model.input, outputs=final_layer)
                                                                                                                                                 ◎ ↑ ↓ 占 ♀ ▮
 [10]:
        efficientnet_model.summary()
```





```
loss_weight = 0.05
efficientnet_model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'],loss_weights = loss_weight)

from tensorflow.keras.callbacks import EarlyStopping

# Set up early stopping to monitor validation accuracy
early_stopping = EarlyStopping(monitor='val_accuracy', patience=3,mode = 'max')

# Train the model with early stopping
efficientnet_model.fit(train_data, epochs=20, validation_data = test_data , callbacks=[early_stopping])
```

## **Model Validation and Evaluation Report (5 marks):**

Model	Summary				Training and Validation Performance Metrics			
<u>_</u>	[14]: 'qgps_nood_sentry()			814471				
	Model: "functional_1"							
	Layer (type)	Output Shape	Paras e					
	input_layer_1 (InputLayer)	(Name, 224, 224, 3)	.0		[31]: model1=vgg16_model			
	blocks_com/1 (Com/20)	(tone, 226, 226, 68)	1,792		from tensorflow.keras.callbacks import EarlyStopping			
	blocki_com/2 (Com/20) blocki_pool (MaxMoolingID)	(None, 226, 226, 64) (None, 112, 112, 64)	35,928		# Set up early stopping to monitor validation accuracy			
	blocks convs (convs)	(NOTE, 112, 112, 128)	71.056		early_stopping = EarlyStopping(monitor='val_accuracy', patience=3,mode = 'max')			
	Black2_com/2 (com/20)	(Name, 112, 112, 128)	247,534		# Train the model with early stopping			
	block2_pool (MaxPoolingID)	(term, 50, 50, 120)	4		<pre>vgg16_model.fit(train_data, epochs=15, validation_data=test_data,callbacks=[early_stopping])</pre>			
0016	block3_convi (Com/20)	(Home, S6, S6, 256)	295,168		Epoch 1/15 144/144 41s 3s/step - accuracy: 0.7522 - loss: 0.0704 - val_accuracy: 0.9171 - val_loss: 0.0244			
GG16	block3_comv2 (Comv20)	(4000, 56, 56, 256)	590,000		Epoch 2/15			
	block3_com/3 (Com/2b)	(None, 56, 56, 256)	550,000		144/144			
	blocks_pool (HaxPoolingID)	(Hone, 25, 28, 256)	0		144/144			
radal	block4_convs (convst)	(0000, 25, 20, 512)	1,100,100		Epoch 4/15 144/144 — 289s 2s/step - accuracy: 8.9489 - loss: 8.8238 - val_accuracy: 8.9379 - val_loss: 8.8272			
lodel	Block4_conv2 (conv20) Block4_conv2 (conv20)	(None, 26, 28, 612)	2,359,086		Epoch 5/15 144/144			
	block4_pool (MaxPoolingID)	(Nome, 54, 56, 512)	2,00,000		Epoch 6/15			
	blockS_convi (Conv20)	(scre, 14, 18, 512)	2,359,686		144/144 293s 2s/step - accuracy: 0.9533 - loss: 0.0221 - val_accuracy: 0.9493 - val_loss: 0.0219 Epoch 7/15			
	blackS_com/2 (Com/20)	(tone, 14, 14, 512)	2,359,686		144/144 301s 2s/step - accuracy: 8.9513 - loss: 8.8231 - val_accuracy: 8.9458 - val_loss: 8.8298			
	blocks_copys (conv20)	(Hore, 14, 14, 512)	2,359,085		Epoch 8/15 144/144 303s 2s/step - accuracy: 0.9517 - loss: 0.0260 - val accuracy: 0.9479 - val loss: 0.0242			
			70					
	blocks_pool (MaxMoolingID)	(tone, 7, 7, 512)			Epoch 9/15			
	Flatten_1 (Flatten)  flatten_1 (Flatten)  deme_1 (Deme)  total perms: 14,815,044 (64.51.48)  frainable parms: 100,364 (32.40 %  non-trainable parms: 14,724,080 (1	(1000, 25800) (1000, 4) )	0 0 380,755					
	flatten_s (flatten)  dense_1 (Dense)  Total peress: 14,815,044 (56.51 AB)  Trainable paress: 300,356 (392.03 B	(1000, 25800) (1000, 4) )	8 100,755		Epoch 9/15 144/144 286s 2s/step - accuracy: 0.9503 - loss: 0.0208 - val_accuracy: 0.9406 - val_loss: 0.0253			
	Gitter_I (Platter)  dest_I (Dome)  field premis (SLII) #6 (SLII #6)  freinole premis (M0,76 (DIA #6)  non-tainable paramo 14,774,680 (1	(1000, 25800) (1000, 4) )	8 8 300,755		Egoch 9/15  144/146  2866 24/4tep - accuracy: 0.9583 - loss: 0.0288 - val_accuracy: 0.9486 - val_less: 0.0253  [B1]: Ckeras.arc.callbocks.history.History et 0/2499/136949  [21]: Free tencerflow.karas.callbacks import EarlyStopping  # Set up early stopping to monitor validation accuracy			
	Flatter_1 (Platter)  Sem_2 (Semin)  Teld press (54.51 Pd)  Trainble press (59.76 (56.51 Pd)  Trainble press (59.76 (56.51 Pd)  To children press (59.76 (56.51 Pd)  To children press (59.76 (56.51 Pd)  Trainble	(1000, 25800) (1000, 4) )	8 8 300,554	○ ↑ ↓ △ 平	Egoch 9/15  144/146  2864 24/48p - accuracy: 0.9500 - loss: 0.0200 - val_accuracy: 0.9406 - val_loss: 0.0213  [81]			
	Gatter, Irlates desag, Chomes held prescribes, fig. 10 held prescribes, fig. 10 held prescribes, fig. 10 held prescribes to the con- nection prescribes to the con- traction of the con- traction of the con- position of the con- the con- traction of the con- the con- traction of the con-	(Cools, 2988) (Cools, 4) ) ) (S) ) (4) ) (4)		0 ↑ ↓ △ ₹	Egoch 9/15 144/146 — 286s 2s/step - accuracy: 0.9503 - loss: 0.0200 - val_accuracy: 0.9406 - val_loss: 0.0223 [B1]: Ckeras.arc.callbacks.history.History et 0x2409f336969-  [B1]: Free tensorflow.karas.callbacks import EarlyStopping # Set up early stopping to monitor validation accuracy			
	Gates, Irlands  Johns, Common  Joseph Permis 19,121,004 (S.S.1 40)  Trinschip permis 19,121,004 (S.S.1 40)  Trinschip permis 19,120,005 (S.S.1 40)  Mon Grandlo permis 19,720,005 (S.S.1 40)  [0] remotification permis 19,720,005 (S.S.1	(const, 2008) (cons, 4) ) (3) (5) (6) (6) (6) (6) (6) (7) (7) (8) (6) (7) (8) (8) (8) (8) (8) (8) (8) (8) (8) (8	BIOLOGIA	<b>电介心点</b> 节	Egoch 9/15  184/184  2866 24/18tp - accuracy: 8,9500 - loss: 8,0206 - val_accuracy: 8,9606 - val_loss: 8,0213  [31]: ceres.src.callbacks.history.History et 0/2409f1369f0  [31]: from tensorFlow.keras.callbacks impart EarlyStopping  = 5.5t up monly stopping to monitor validation accuracy aerly_stopping is EarlyStopping monitors val_accuracy', patiences)  # Trust the model wife monity stopping remott0_model.rift rain_datas, apochas20, validation_datastest_data, callbacks:[marly_stopping])  Epoch 1/20			
	Gatter, Irlates dest	(Cools, 2988) (Cools, 4) ) ) (S) ) (4) ) (4)		<b>电子心点节</b>	Egoch 9/15  184/184  2866 24/18tp - accuracy: 8,9500 - loss: 8,0206 - val_accuracy: 8,9606 - val_loss: 8,0213  [31]: ceres.src.callbacks.history.History et 0/2409f1369f0  [31]: from tensorFlow.keras.callbacks impart EarlyStopping  = 5.5t up monly stopping to monitor validation accuracy aerly_stopping is EarlyStopping monitors val_accuracy', patiences)  # Trust the model wife monity stopping remott0_model.rift rain_datas, apochas20, validation_datastest_data, callbacks:[marly_stopping])  Epoch 1/20			
	Gattery, Visited  detail (Visited)  Test appears (**)  Tes	(Cools, 2588) (Cools, 4) (Cools, 4) (Cools, 4) (Cools, 4) (Cools, 4) (Cools, 4) (Cools, 5)	Perm I Connected to	0 ^ + A T	Egoch 9/15  144/144  2864 24/48p - accuracy: 0.9500 - loss: 0.0200 - val_accuracy: 0.9406 - val_loss: 0.0223  [81] Ceres.arc.callbacks.history.History et 0/2409f3369f0-  [82] from tensorflow.keras.callbacks import EarlyStopping  # Set up early stopping to monitor validation accuracy early.topping = EarlyStopping(monitors'val_accuracy; patiences) # Froit the model drie monity stopping resential_model.rist(rain_dats, uponhaze), validation_datest_dats_callbacks:[early_stopping]    Epoch 12/30   144/144			
	Gattery, Visited  detail (Visited)  Test appears (**)  Tes	Concy, 28800 (100; 40) (10	Perms 7   Connected to   0   0   0   0   0   0   0   0   0	0 + + 4 7	Egoch 9/15  144/144  231s 2/step - accuracy: 8.9580 - loss: 8.6228 - val_accuracy: 8.9486 - val_loss: 8.6223  [21]: deers.src.callbacks.history.History et 0/2499f136969.  [21]: from tensorflow.kers.callbacks impart EarlyStopping  # Set up morely stopping to monitor validation accuracy early_stopping = EarlyStopping(monitor*val_accuracy*, patiences))  remott0_model.fit(train_data, spochs:20, validation_datastest_data, callbacks:[serly_stopping])  Egoch 1/20  144/144  231s 2/step - accuracy: 0.8584 - loss: 0.0266 - val_accuracy: 0.9394 - val_loss: 0.005  144/144  231s 2/step - accuracy: 0.9489 - loss: 0.0066 - val_accuracy: 0.9397 - val_loss: 0.055  144/144  237s 2/step - accuracy: 0.9489 - loss: 0.0066 - val_accuracy: 0.921 - val_loss: 0.055  144/144  237s 2/step - accuracy: 0.9489 - loss: 0.0066 - val_accuracy: 0.921 - val_loss: 0.055			
	Gatter_i (Taiter_i)  Jenin_i (Senso)  Jenin_i (Senso)  Jenin_i (Senso)  Jenin_i (Senso)  Trainable parent: 100,700 (200.0)  Inc. Grainable parent: 10,700,600 (1  Trainable parent: 10,700,600 (1  Jenin-Grainable pa	Concy, 28800 (100; 40) (10	Parent F Connected to 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 - 5 -	0 1 4 A V	Egoch 9/15  144/146  2864 24/4tep - accuracy: 0.9503 - loss: 0.0208 - val_accuracy: 0.9486 - val_loss: 0.0203  [01] Ceress.arc.callbacks.history.History et 0/2409f136969.  [21] From tensorflow.keras.callbacks import EarlyStopping  # Set up early stopping to monifer validation accuracy aerly_stopping familyStopping familyStopping of the sold of the sold with early stopping familyses are ly_stopping familyses validation_datastest_data, callbacks:[early_stopping])  [22] Front the model with early stopping familyses are large family accuracy: 0.0004 - loss: 0.0206 - val_accuracy: 0.9514 - val_loss: 0.0051  [24] M1/444  [25] M4/244  [26] M4/244  [27] M4/244  [28] M4/26  [28] M4/26  [28] M4/27  [29] M4/27  [29] M4/28  [20] M3/28 - accuracy: 0.9507 - loss: 0.0006 - val_accuracy: 0.907 - val_loss: 0.0016  [20] M4/27  [20] M3/28 - accuracy: 0.9507 - loss: 0.0008 - val_accuracy: 0.9407 - val_accuracy: 0.9417 - val_loss: 0.0116  [20] M4/27  [20] M4/28 - M4/28  [20] M4/28 - B4/28  [2			
	Gattery, Tribites  desar, Tommer)  Test present (**pitt), bet (b.t.) ret  Testable present (**pitt), bet (b.t.) ret  Testable present (**pitt), bet (b.t.) ret  Testable present (**pitt), bet (b.t.) ret  [7]	Cone, 286, 286, 31  Cone, 45  Cone, 15, 286, 286, 31  Cone, 286, 386, 386, 386	Perm #   Connected to	0 1 2 4 7	Egoch 9/15  184/184  2866 24/1869 - accuracy: 0.9500 - loss: 0.0208 - val_accuracy: 0.9406 - val_loss: 0.0203  [31]			
	Gatter_i (Triation)  Joint_i (Source)  Joint_i (	Conc., 2880 (100; 44)  (100; 44)  (100; 45)	Parax	0 ↑ ↓ △ ♥	Egoch 9/15   2864 24/step = accuracy: 0.9500 - loss: 0.0208 - val_accuracy: 0.9486 - val_loss: 0.0203			
esnet-50	Gattery, Irlates)  destar, Inmen)  held press: hydrighes that not  held press: hydrighes that not  me-trained parent: hydrighes  pen-trained parent: hydrighes  (1)  resortE_midst.temmey()  not1: "nextine"  Laser (type)  Laser (type)  cont_pan (Correlation)  cont_pan (Correlation)  cont_pan (Correlation)  pent_pan (Correlation)	Cone, 2880 (1994 Nove Cone, 124, 124, 44) (1994 Nove, 124, 124, 124, 44) (1994 Nove, 124, 124, 124, 124, 124, 124, 124, 124	Ferms	9 1 4 4 7	Egoch 9/15   2864 24/step = accuracy: 0.9500 - loss: 0.0208 - val_accuracy: 0.9486 - val_loss: 0.0203			
esnet-50	Gatter_i (Triation)  Joint_i (Source)  Joint_i (	Conc., 2880 (100; 44)  (100; 44)  (100; 45)	Parax	日子かる节	Egoch 9/15   2864 24/step = accuracy; 0.9503 - loss: 0.0208 - val_accuracy; 0.9486 - val_loss: 0.0203			
esnet-50	Gattery, Tribition  Joint, Tommon)  Test present (spill), the US-Li rel  Testable present (spill)  Testable present (spil	Cone, 2880 (1994 Nove Cone, 124, 124, 44) (1994 Nove, 124, 124, 124, 44) (1994 Nove, 124, 124, 124, 124, 124, 124, 124, 124	Ferms	0 1 4 A T	Egoch 9/15   2864 24/14ep - accuracy: 0.950 - loss: 0.0208 - val_accuracy: 0.9486 - val_loss: 0.0233			
esnet-50	Gattery, Tribition  Joint, Tommon)  Test present (spill), the US-Li rel  Testable present (spill)  Testable present (spil	(1000, 3800) (1000, 4) (10	Primal E   Collected to	0 ↑ ÷ & ∀	Egoch 9/15   2866 24/step - accuracy: 0.9500 - loss: 0.0208 - val_accuracy: 0.9405 - val_loss: 0.0223			
esnet-50	Gatter, Italies  desta_1 (Deste)  Test press (Application)  Test press (Application)  Test press (Application)  Test press (Application)  Test (Application)  Test (Application)  Test (Test (Test (Application))  Test (Test (Test (Application)))  Test (Test (	Conc., 2800 (Conc., 40)  (Conc., 40)  (Conc., 40)  (Conc., 40)  (Conc., 20)  (Conc.	Paris	0 + + 4 7	Egoch 9/15   2864 24/14Fp - accuracy; 0.9503 - loss: 0.0208 - val_accuracy; 0.9406 - val_loss: 0.0223			
esnet-50	Gattery, Tribition  desiry, Tribition  being recent logistics (bill rec  being recent logistics (bill rec  being recent logistics (bill rec  being recent logistics (bill recent logistics)  [7] recently, middl.temmey()  book! "factions"  Laper (type)  lame_laper (technique)  conclusion (techniq	Cone, 2800 (100 (100 (100 (100 (100 (100 (100 (	Parent	0 1 4 A T	Egoch 9/15   184/154   2864 24/158p - accuracy; 8.9583 - loss: 8.0298 - val_accuracy; 8.9586 - val_loss: 8.0253			
esnet-50	Gatters, l'infatteno  Josins, l'Ommen)  Total permit 14,511,504 (S.S.1 40)  Totales permit 19,511,504 (S.S.1 40)  Totales permit 19,511,504 (S.S.1 40)  Totales permit 19,511,504 (S.S.1 40)  Totales permit 15,511,504 (S.S.1 40)  Totales permit 15,511,511,511,511,511,511,511,511,511,	(100%, 4) (100%,	Premi	0 + ÷ ± T	Egoch 9/15   184/154   2864 21/156p - accuracy; 8.9503 - loss: 8.0208 - val_accuracy; 8.9406 - val_loss: 8.0203     [31]			
esnet-50	Gatter, Litation  seast_ (Dome)  Tell press (seaster)  Tell press	Comp., 2000 (1000, 40 )  (2000, 40 )  (200, 40 )  (200, 40 )  (200, 40 )  (200, 40 )  (200, 104, 104 )  (200, 104 )  (200	Paras I Connected to  9	0 + 4 a T	Egoch 9/15   184/154   2864 24/158p - accuracy; 8,9583 - loss: 8.0298 - val_accuracy; 8,9486 - val_loss: 8.0253			
Resnet-50	Gattery, Tribition  Joseph press 1-y-115,000 (Sc.31 re)  Total press 1-y-115,000 (Sc.31 re)  Pool. 1-y	Concy, 2000 (1000	Parent	0 * + 4 *	Egoch 9/15   2864 21/step - accuracy; 0.9500 - loss: 0.0208 - val_accuracy; 0.9406 - val_loss: 0.0253     [31]   Geres.src.callbacks.history.History at 0.020971309709.       [31]   From tensorflow.karsa.callbacks.impert EurlyStopping			
esnet-50	Gatters, Linkstein  Johns, Domen  John J. (2000)  John J. (200	(2009, 40) (30) (40) (40) (40) (40) (40) (40) (40) (4	Paren	0++47	Egoch 9/15   184/154   2864 21/156p - accuracy; 0.9503 - loss: 0.0208 - val_accuracy; 0.9456 - val_loss: 0.0233   [31]:			
esnet-50	Gatters, Linkstein  Johns, Domen  John J. (2000)  John J. (200	Concy, 2000 (1000	Parent	0 1 4 A V	Egoch 9/15   184/144   799 5/14tep - accuracy; 0.955 - loss: 0.0258 - va2_accuracy; 0.9466 - va2_loss: 0.0233   [31]   ceres.irc.callsecs.history_History_at 0.0409f136969.			





conv2_block3_2_relu (Activation)	(None, 56, 56, 64)		conv2_block3_2_bn[e][e]	
conv2_block3_3_conv (Conv20)	(None, S6, S6, 2S6)	16,640	conv2_block3_2_relu(0)(0)	
conv2_block3_3_bm (BatchNormalization)	(Mone, 56, 56, 256)	1,024	conv2_block3_3_conv(e)(e)	
conv2_block3_add (Add)	(Mone, 56, 56, 256)		conv2_block2_out[e][e], conv2_block3_3_bn[e][e]	
conv2_block3_out (Activation)	(Mone, 56, 56, 256)	0	conv2_block3_add[e][e]	
canv3_block1_1_canv (conv20)	(Mone, 28, 28, 128)	32,896	conv2_block3_out[e][e]	
conv2_block1_1_bn (BatchHormalization)	(Mone, 28, 28, 128)	512	conva_block1_1_conv[e][e]	
conv3_blocki_i_relu (Activation)	(None, 28, 28, 128)		comv3_block1_i_bn(e)(e)	
conv3_block1_2_conv (conv20)	(None, 28, 28, 128)	147,584	convd_block1_1_relu[0][0]	
conv3_block1_2_bn (SetchNormelization)	(Mone, 28, 28, 128)	512	convs_blocks_s_conv[e][e]	
conv3_block1_2_relu (Activation)	(None, 28, 28, 128)		conv3_block1_2_bn(e)(e)	
conv3_block1_0_conv (Conv20)	(None, 28, 28, 512)	131,584	conv2_block3_out[e][e]	
conv3_blocki_3_conv (Conv10)	(None, 28, 28, 512)	66,848	conv3_blocki_2_relu[0][0]	
conv3_blocki_0_bn (matchwormalization)	(None, 28, 28, 512)	2,048	conv3_block1_0_conv[0](0]	
conva_blocks_a_bn (BatchNormalization)	(Mone, 28, 28, 512)	2,048	conva_blocks_s_conv[e](e)	
conv3_blocki_edd (Add)	(tone, 28, 28, 512)		comv3_block1_0_bn[0][0], comv3_block1_3_bn[0][0]	
conva_blocks_out (Activation)	(Mone, 28, 28, 512)		conv3_block1_add[e][e]	
conv3_block2_1_conv (Conv20)	(Mone, 28, 28, 128)	65,664	conv3_block1_out[e][e]	
conv3_block2_1_bn (BatchNormalization)	(Mone, 28, 28, 128)	512	conv3_block2_1_conv[e][e]	
conv3_block2_1_relu (Activation)	(Mone, 28, 28, 128)		conv3_block2_1_bn(e)(e)	
canv3_block2_2_canv (Conv20)	(None, 28, 28, 128)	147,584	conv3_block2_i_relu[0][0]	
conv3_block2_2_bm (matchwormalization)	(Mone, 28, 28, 128)	512	conv3_block2_2_conv[0][0]	
conva_block2_2_relu	(Mone, 28, 28, 128)	•	conva_block2_2_bn[e][e]	

convs_block2_1_relu (Activetion)	(sone, 7, 7, 512)		convs_block2_1_bn(e)(e)	
conv5_block2_2_conv (Conv20)	(sone, 7, 7, 512)	2,359,886	conv5_block2_i_relu[#][#]	
conv5_block2_2_bn (BatchNormalization)	(Marrie, 7, 7, 512)	2,048	conv5_block2_2_conv[0][0]	
conv5_block2_2_relu (Activation)	(Mone, 7, 7, 512)		conv5_block2_2_bn[#][#]	
convs_block2_3_conv (Conv20)	(None, 7, 7, 2048)	1,050,624	convs_block2_2_relu[e][e]	
convs_block2_s_bn (BetchNormelization)	(None, 7, 7, 2046)	8,192	convs_block2_1_conv[e][e]	
conv5_block2_add (Add)	(None, 7, 7, 2048)		comv5_block1_out[e][e], comv5_block2_3_bn[e][e]	
conv5_block2_out (Activation)	(Sone, 7, 7, 2848)		conv5_block2_add[e][e]	
conv5_block3_i_conv (Conv20)	(sone, 7, 7, 512)	1,049,068	conv5_block2_out[e][e]	
conv5_block3_i_bn (Batchwormalization)	(None, 7, 7, 512)	2,048	conv5_block3_s_conv[#][#]	
convs_blocks_i_relu (Activation)	(None, 7, 7, 512)		convs_blocks_s_bn[e][e]	
conv5_block3_2_conv (Conv20)	(None, 7, 7, 512)	2,359,888	convs_block3_1_relu(0)[0]	
conv5_block3_2_bn (BatchNormalization)	(None, 7, 7, 512)	2,048	convs_block3_2_conv(e)(e)	
conv5_block3_2_relu (Activation)	(None, 7, 7, 512)		conv5_block3_2_bn[#][#]	
convs_blocks_s_conv (Conv20)	(sone, 7, 7, 2048)	1,050,624	convs_blocks_2_relu[e][e]	
conv5_block3_3_bn (BatchNormalization)	(None, 7, 7, 2848)	8,192	conv5_block3_3_conv[e][e]	
conv5_block3_add (Add)	(None, 7, 7, 2046)		conv5_block2_out[e][e], conv5_block3_3_bn[e][e]	
conv5_block3_out (Activation)	(sone, 7, 7, 2048)		conv5_block3_add[e][e]	
flatten (Flatten)	(None, 100352)		com/5_block3_out[0][0]	
dense (Dense)	(sone, 4)	401,412	flatten[0][0]	

Trainable params: 401,412 (1.53 MB)
Non-trainable params: 23,587,712 (89.98 MB

# Efficient Net

Layer (type)	Output shape	Paran #	connected to	
input_layer (inputLayer)	(Mone, 224, 224, 3)			
rescaling (mescaling)	(Mone, 224, 224, 3)		input_layer[e][e]	
normalization (Normalization)	(sone, 224, 224, 3)	7	rescaling[e][e]	
rescaling_1 (Mescaling)	(Mone, 224, 224, 3)		normalization(e)[e]	
stem_conv_pad (ZeroFadding2D)	(sone, 225, 225, 3)		rescaling_1[e][e]	
stem_conv (Conv20)	(mone, 112, 112, 32)	864	stem_corv_pad(e)[e]	
stem_bn (matchwormalization)	(Mone, 112, 112, 12)	128	stem_conv[e](e)	
stem_activation (activation)	(Mone, 112, 112, 32)		stem_bn(e)(e)	
blockia_dwconv (DepthwiseConv20)	(None, 112, 112, 32)	288	sten_activation(e)(e)	
blockia_bn (Batchwormalization)	(None, 112, 112, 32)	128	blockia_duconv[e][e]	
blockia_activation (Activation)	(More, 112, 112, 32)	•	blockia_bn(@)(@)	
blockia_se_squeeze (GlobalAveragePooling20)	(None, 32)		blockis_activation[0][0]	
blockIa_se_reshape (Reshape)	(None, 1, 1, 32)		blockla_se_squeeze(0)(0)	
block1a_se_reduce (Conv2D)	(mone, 1, 1, 8)	264	blockla_se_reshape(0)(0)	
blockla_se_expand (Conv2D)	(None, 1, 1, 32)	288	block1a_se_reduce[0](0)	
block1a_se_excite (Multiply)	(None, 112, 112, 32)	•	blockia_activation(e)(e), blockia_se_expand(e)(e)	
blockia_project_comv (Comv2D)	(Mone, 112, 112, 16)	512	blockia_se_excite[e][e]	
blockia_project_bn (EstchWormalization)	(None, 112, 112, 16)	64	blockia_project_conv[0][0]	
block2a_expand_comv (Conv20)	(None, 112, 112, 96)	1,536	block1a_project_bn(0)(0)	
block2a_expand_bn (SatchWormalization)	(None, 112, 112, 96)	384	block2a_expand_conv[0](0)	
block2a_expand_activation (Activation)	(Mone, 112, 112, 96)		block2a_expand_bn[e][e]	





