Logo

Description automatically generated

**Ain shams university**

**Faculty of computer and information science**

**Bioinformatics department**

**Ain Shams University**

**Faculty of Computer and Information Science**

**Scientific Computing department**

**Project Title**

**Handwritten Signature Identification and Verification**

|  |  |  |
| --- | --- | --- |
| **Name** | **ID** | **Section** |
| **Ahmed Magdy Abdelrahman** | **20191700056** | **1** |
| **Abdul-Rahman Sayed Ali Mohammed** | **20191700339** | **3** |
| **Sara Mohammed Osman** | **20191700279** | **3** |
| **Mohammed Nour-Elden Abbas Ismael** | **20191700583** | **4** |
| **Nour Mohamed Hussein Kamaly** | **20191700701** | **5** |

**Under the supervision of**

**Dr. Dina Khattab**

**Scientific Computing Department,**

**Faculty of Computer and Information Science**

**Ain Shams University**

**Data Preparation**

**- There are 200 training images (40\*5) with 5 classes as output: PersonA, PersonB, PersonC, PersonD, and PersonE.**

**- We applied Augmentation to increase the dataset, we now have 991 training images (198\*5) and 200 validation images (40\*5) with 5 classes as output: PersonA, PersonB, PersonC, PersonD, and PersonE.**

**- Concatenate the training and test sets from the 5 folders.**

**- We tried to use gans model to create images but not enough efficient to use it.**

**Description of the models and techniques.**

**ResNet-50:**

**- Input shape:**

**input\_shape = (224, 224, 3), include\_top = False, weights = 'imagenet'**

**- Add some layers which match the dataset:**

**x = layers.Flatten()(base\_model.output)**

**x = layers.Dense(512, activation='relu')(x)**

**x = layers.Dropout(0.5)(x)**

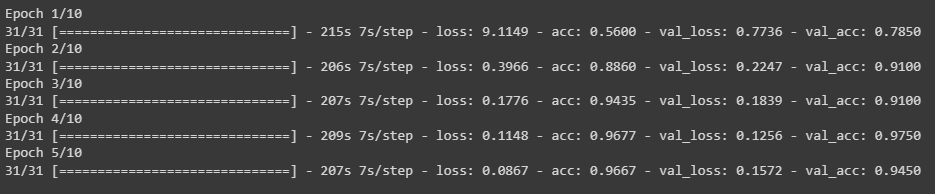
**x = layers.Dense(5, activation='Softmax')(x)**

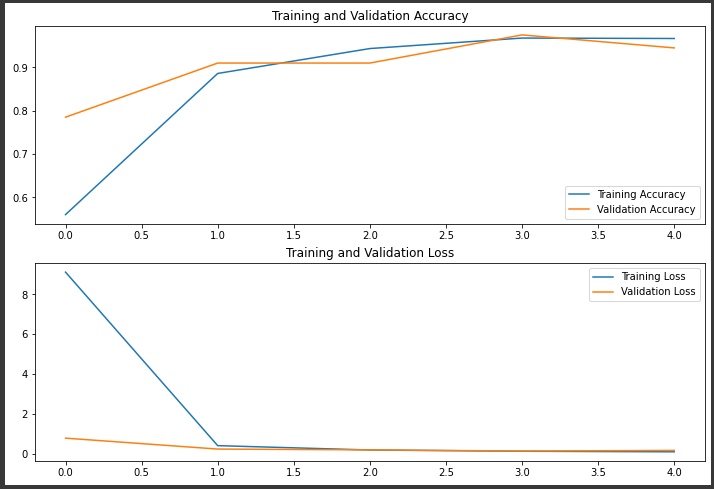
**model = tf.keras.models.Model(base\_model.input, x)**

**- Optimizer & Loss:**

**optimizer = 'adam', loss = 'categorical\_crossentropy'**

**- Fit Model:**

****

**- History Plot:**

**Inception-V3:**

**- Input shape:**

**input\_shape = (150, 150, 3), include\_top = False, weights = 'imagenet'**

**- Add some layers which match the dataset:**

**x = layers.Flatten()(base\_model.output)**

**x = layers.Dense(512, activation='relu')(x)**

**x = layers.Dropout(0.2)(x)**

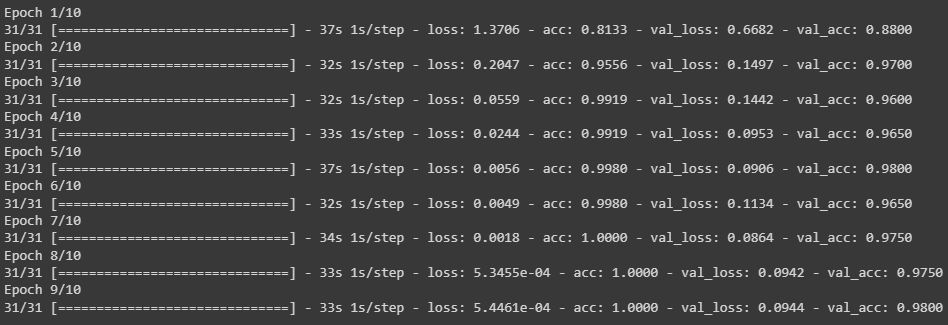
**x = layers.Dense(5, activation='Softmax')(x)**

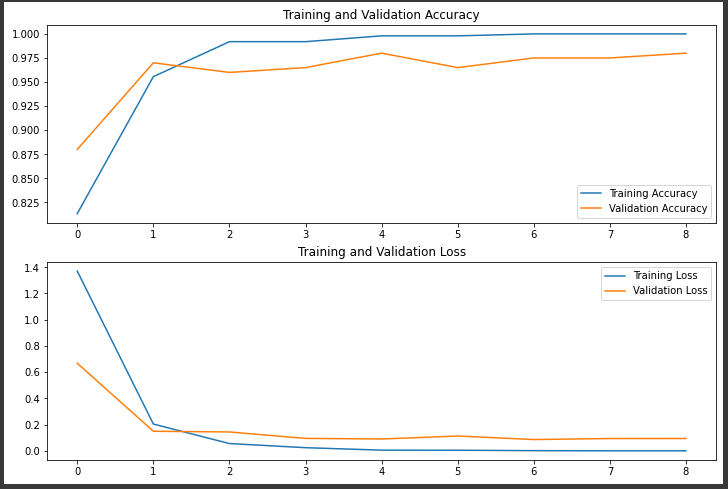
**model = tf.keras.models.Model(base\_model.input, x)**

**- Optimizer & Loss:**

**optimizer = 'adam', loss = 'categorical\_crossentropy'**

**- Fit Model:**

****

**- History Plot:**

**VGG-16:**

**- Input shape:**

**input\_shape = (224, 224, 3), include\_top = False, weights = 'imagenet'**

**- Add some layers which match the dataset:**

**x = layers.Flatten()(base\_model.output)**

**x = layers.Dense(512, activation='relu')(x)**

**x = layers.Dropout(0.5)(x)**

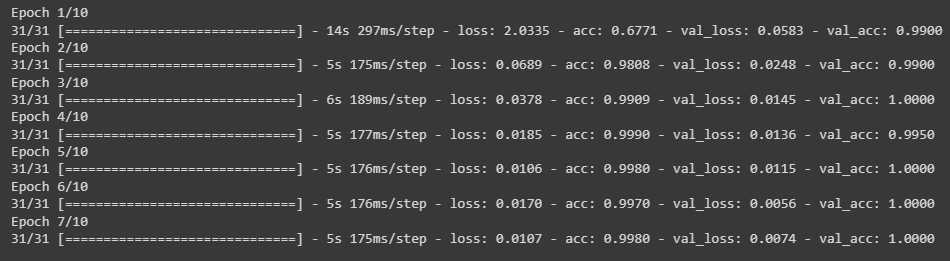
**x = layers.Dense(5, activation='Softmax')(x)**

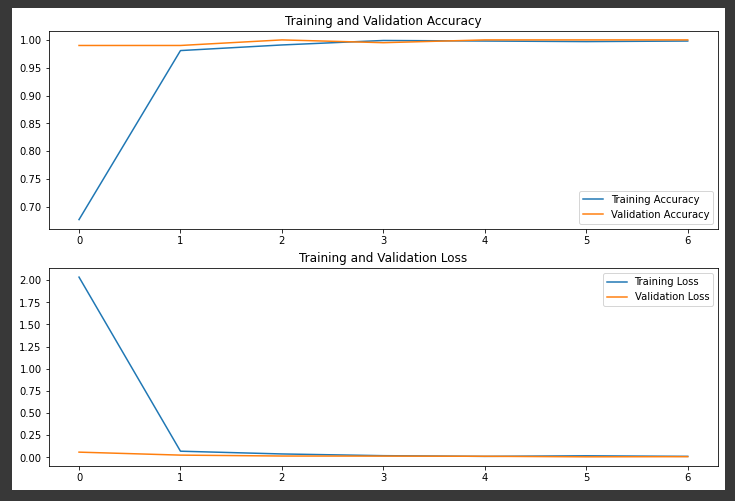
**model = tf.keras.models.Model(base\_model.input, x)**

**- Optimizer & Loss:**

**optimizer = 'adam', loss = 'categorical\_crossentropy'**

**- Fit Model:**

****

**- History Plot:**

**Vision Transformers:**

**- Input shape: input shape =(200,200,3)**

**- Add some layers which match the dataset:**

**Diagram

Description automatically generated with medium confidence**

**- Optimizer & Loss: loss function is cross entropy and optimizer is adam with parameters lr=0.001, weight\_decay=0.0001**

**Bag Of Words (BOW):**

**- Input shape:**

**- Add some layers which match the dataset:**

**- Optimizer & Loss:**

**- Fit Model:**

**- History Plot:**

**Siamese:**

**- Input shape:**

**- Add some layers which match the dataset:**

**- Optimizer & Loss:**

**- Fit Model:**

**- History Plot:**

**Object Detection using YOLO:**

**- CSV handling for training:**

Table

Description automatically generated

**- Input shape: 1000,1000,3**

**- Add some layers which match the dataset:**

**Table

Description automatically generated**

**- Optimizer & Loss:**

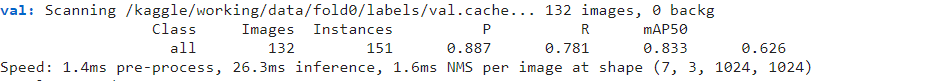
****

**- Fit Model:**

**A picture containing table

Description automatically generated**

**- History Plot:**



**Graphical user interface

Description automatically generated**

**- Detection sample:**

Graphical user interface, text, application

Description automatically generated**Graphical user interface, text, application

Description automatically generated**