IIPRD

KHURANA & KHURANA
ADVOCATES AND IP ATTORNEYS

E-13, UPSIDC Site-IV, Kasna Road, Greater Noida - 201308, India

Phone: +91.120.4296878 | Website: www.khuranaandkhurana.com www.iiprd.com
Email: info@khuranaandkhuraana.com | iiprd@iiprd.com
Indian Offices: Noida | New Delhi | Bangalore | Mumbai | Pune | Mumbai | Hyderabad |Jalandhar | Chennai
International Offices: US |Bangladesh | Sri Lanka | Malaysia | Vietnam | Myanmar | Nepal | Malaysia

INVENTIVE DISCLOSURE - CONFIDENTIAL

1. **Proposed Title of the Invention** (Not more than 10 to15 words)

Infant Health Guard Tracking with "YOSO" Speech Detection & Face Expression Recognition.

2. **Proposed Abstract of the Invention** (Kindly explain the crux of the invention in about150 to 200 words.)

This paper presents a novel infant health monitoring system designed specifically for babies in incubators, where the presence of multiple infants makes regular monitoring by hospital staff challenging. The key innovation of this project lies in enhancing baby security by distinguishing between strangers and relatives near the crib. This is achieved through the implementation of the YOSO speech algorithm, which involves creating a dataset comprising voices of individuals related and unrelated to the baby to ensure protection. We have created our own infants sample datasets and that contributes to our novelty.

The study explores the application of facial recognition techniques in this context, involving the compilation and pre-processing of a varied dataset of infant facial images. Features like facial landmarks and color distribution are extracted from these images, and machine learning methods, particularly Convolution neural networks (CNNs), are employed to train and test a model for better accuracy. Also there is a Stranger alarm which notifies whether the baby in the incubator has been carried away a certain distance.This alarm works with the integration of Esp32 Microcontroller.

3. Key Words:

YOSO(You Speak Only Once)
CNN (Convolution NeuralNetwork)
Stranger alarm

4. Background of the Invention:

What are the present technologies that exist in the field of your invention and what are the limitations of the same? (Present state of Art)

Keeping an Eye (and Ear) on the Littlest Ones: Where Infant Monitoring Stands Today There are many ways to keep an eye on your baby these days, from classic video and audio monitors to fancy smart cribs. But each approach has its drawbacks.

Traditional monitors let you see and hear your baby, but they can't tell you much about their health. Wearable devices track heart rate and breathing, but they might be uncomfortable and don't show emotions. Smart cribs offer insights into sleep patterns and movement, but miss out on facial expressions and who's talking to the baby. Even voice recognition tech in monitors can be basic, just detecting cries without recognising who's calming the baby.

The problem Most monitors don't combine these features for a more complete picture. They can't give parents a full understanding of their baby's well-being and safety.Keeping Tabs on Your Tiny Human:

Today, we have a variety of technologies offering a window into their world:

- **Classic monitors:** See & hear baby, but miss health and emotions.
- **Wearable sensors:** Track vitals, but uncomfortable and lack emotion data.
- **Smart cribs:** Monitor movement, sleep, and environment, but miss emotions and speaker ID.
- **Basic speech recognition:** Detects cries, but may not tell who's talking to the baby.

5. **What problems does the invention address and how your Invention is able to overcome the limitations/ problems of the existing technologies?**

Problems such as infant kidnapping happen due to the lack of safety in the incubators. This YOSO technology provides a mechanism to distinguish between relatives and strangers.

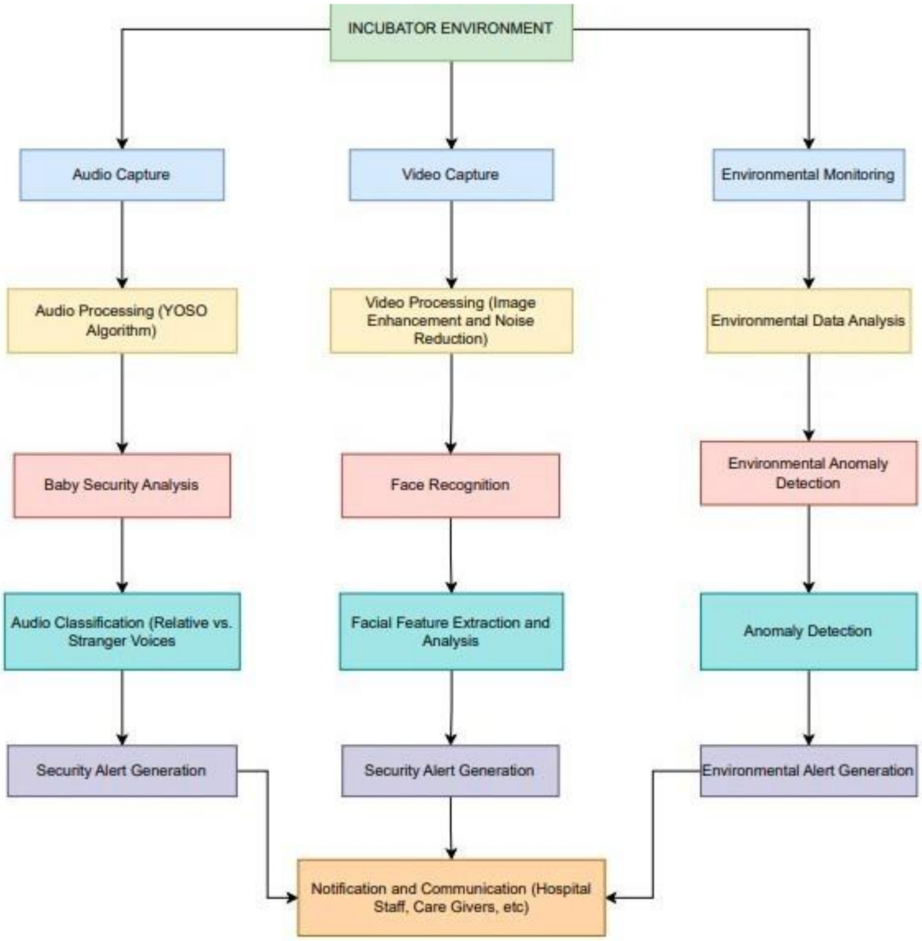
It is difficult to monitor vast number of babies at the same time so proper healthcare cannot be provided at times. To tackle this problem, facial recognition of the infants can be used for monitoring the babies effectively. For facial recognition, we have used CNN(Convolutional Neural Network) to recognize the emotions of the baby such as happy,normal,sleepy,crying. With the integration of YOSO technology, we have built a stranger alarm that senses and raises an alert if a stranger is near the baby ‘s crib. This will tackle the problem of kidnapping.

6. **Detailed Explanation of the Invention along with working examples. Kindly provide an elaborated description of each and every aspect of the invention (product and/or process) in great detail.**

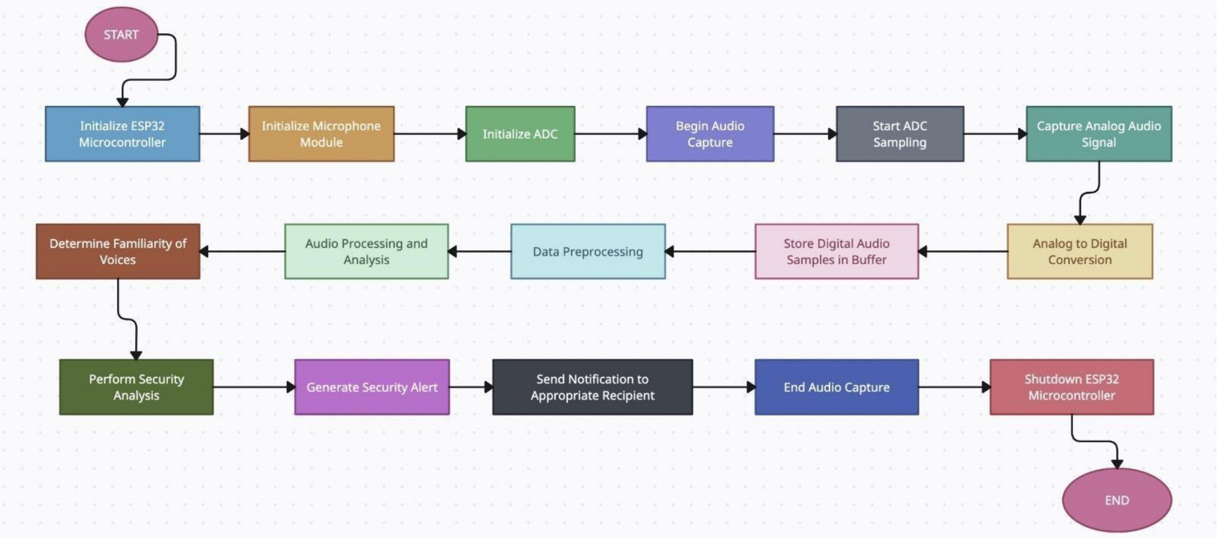
The invention is an advanced infant health guard system installed at incubators in healthcare systems (Fig. 1) ; While strangers pass through, the system employs especially sensitive sensors to detect whether the person nearing the crib is a stranger or a relative. This it does with the implementation of YOSO speech detection algorithm that works on the dataset containing voice samples of the relatives. This technology is enough for the stranger alarm to raise an alert if a stranger tries to invade the baby ‘s crib. The stranger alarm works with the sensor that works with the integration of ESP32 Microcontroller. It’s difficult to monitor each infant individually for the entire day and cater to their needs effectively, so this has been solved by the facial recognition mechanism using CNN. The dataset is just not any random dataset from some open-sources rather than real-time images of various infants. It can effectively recognizes emotions of the baby such as crying,sleeping,happy,normal. This real-time detection and alerting processes enable fast intervention, encouraging a reactive attitude in maintaining an safe and secure environment for the infants in the incubators.

7. **Kindly attach drawings, reports, papers, charts or other materials that may aidin your description.**

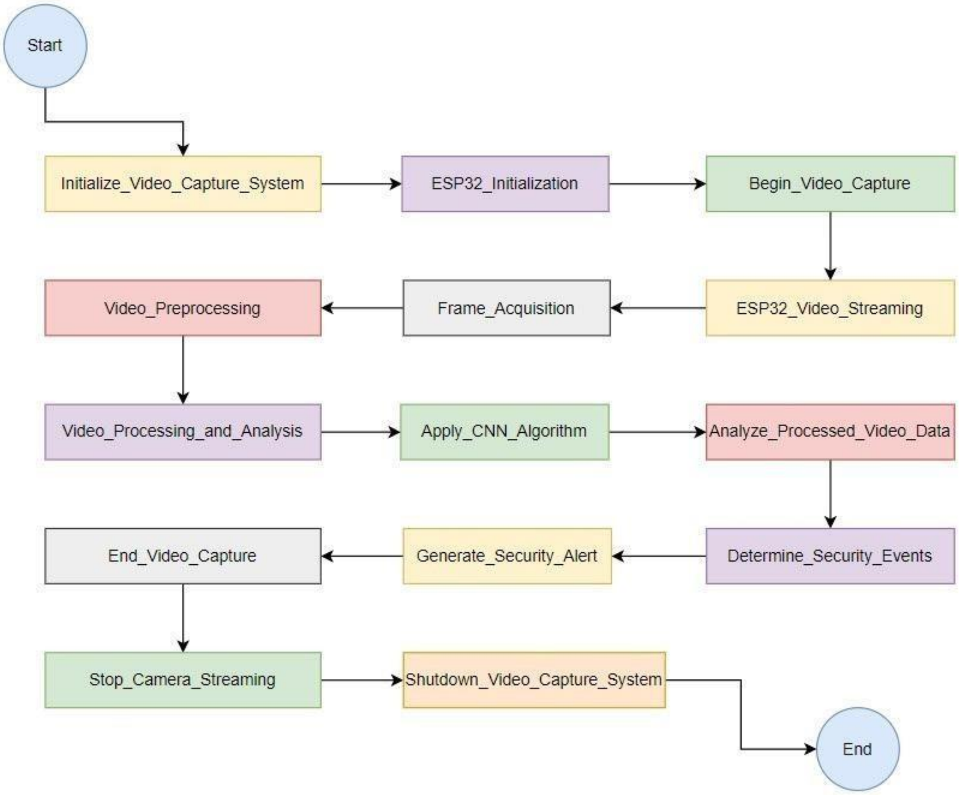
Main flowchart :Fig1


INCUBATOR ENVIRONMENT
Audio Capture
Video Capture
Environmental Monitoring
Audio Processing (YOSO Algorithm)
Video Processing (Image Enhancement and Noise Reduction)
Environmental Data Analysis
Baby Security Analysis
Face Recognition
Environmental Anomaly Detection
Audio Classification (Relative vs. Stranger Voices
Facial Feature Extraction and Analysis
Anomaly Detection
Security Alert Generation
Security Alert Generation
Environmental Alert Generation
Notification and Communication (Hospital Staff, Care Givers, etc)

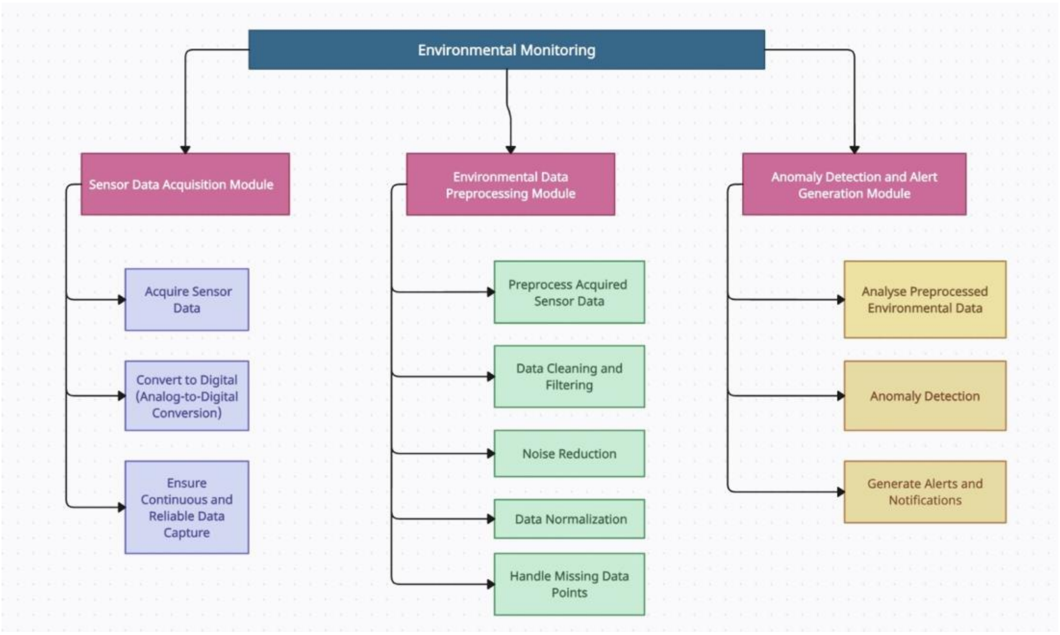
Audio capture sub diagram:

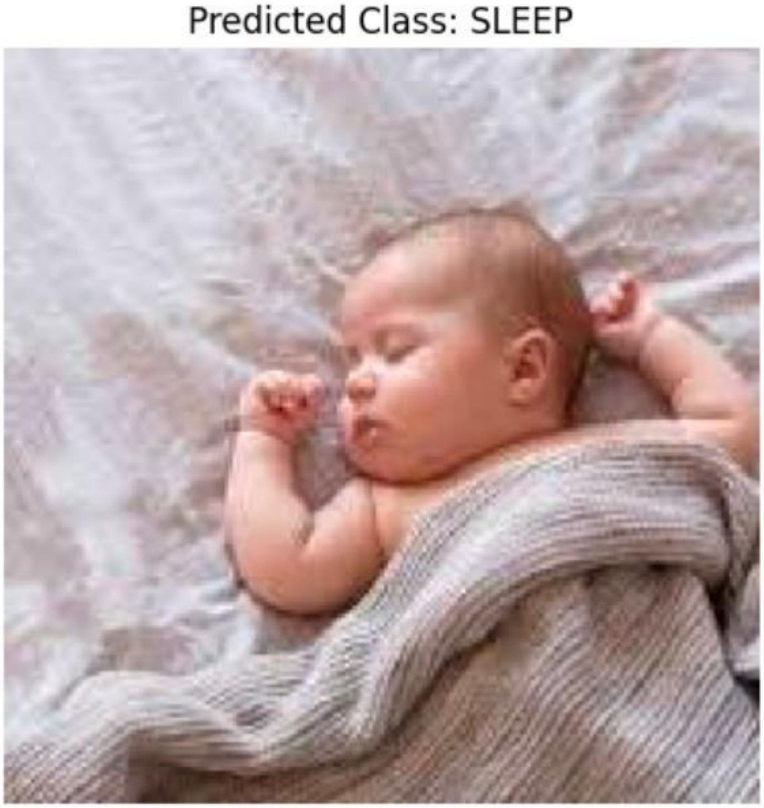

START
Initialize ESP32 Microcontroller
Initialize Microphone Module
Initialize ADC
Begin Audio Capture
Start ADC Sampling
Capture Analog Audio Signal
Analog to Digital Conversion
Store Digital Audio Samples in Buffer
Data Preprocessing
Audio Processing and Analysis
Determine Familiarity of Voices
Perform Security Analysis
Generate Security Alert
Send Notification to Appropriate Recipient
End Audio Capture
Shutdown ESP32 Microcontroller
END

Video capture sub diagram:



Environmental Monitoring sub diagram:


Environmental Monitoring
Sensor Data Acquisition Module
Acquire Sensor Data
Convert to Digital (Analog-to-Digital Conversion)
Ensure Continuous and Reliable Data Capture
Environmental Data Preprocessing Module
Preprocess Acquired Sensor Data
Data Cleaning and Filtering
Noise Reduction
Data Normalization
Handle Missing Data Points
Anomaly Detection and Alert Generation Module
Analyse Preprocessed Environmental Data
Anomaly Detection
Generate Alerts and Notifications



8. **What are the aspects of your disclosure that you want to claim/monopolize?**

   The search was carried out by the use of databases such as scientific journals, conference proceedings, patent databases and internet sources. Its results recoded it serves as a living proof of the novelty of our discovery. The main model we use is a type of convolutional neural network (CNN) that has been previously trained on the data obtained from the Upgraded Public Health Center (UPHC) using an ethical approach. It serves as the center of our plan. YOLO classification model and facial emotion detection are the stand-alone model developed under our approach, contributing to make the system stronger and more user-friendly. Using some of the new technology (combining) has no limits; we destroy the limits and provide top-notch solution for areas such as computer vision and deep learning. The voice recordings added from parents and caretakers considerably improves the accuracy and relevance of our invention, which makes it specifically more effective and useful for yoso detection.

9. **Have you conducted novelty/inventiveness search for your invention? If yes, what are the databases /references used by you? What are the search results?**

YOLO (You Only Look Once) based classification and facial expression recognition under one unified system mark an innovative approach of exploiting deep learning.

The implementation of annotated data, coined by Roboflow, which is carefully prepared for training of models to gain high accuracy and performance.

The availability of the dataset acquired from UPHC with an exclusive access to protect the data source from ethical and lawful use.

10. **Do you feel that a person of “average” skill (not-extraordinary skill) in your areaof technology would have arrived at your invention with existing knowledge in public domain? If no, what could be the reasons for the same?**

It is unlikely that a person of "average" skill in this area would have arrived at this invention with existing knowledge in the public domain. The reasons for this include the complexity and specificity of the solution, which requires a deep understanding of speech algorithms and facial recognition techniques, as well as expertise in machine learning methods.

Furthermore, the need for a comprehensive dataset and the preprocessing involved in extracting meaningful features from infant facial images contribute to the uniqueness of this invention. Therefore, it can be considered a significant advancement in the field of infant health monitoring.

11. **Kindly provide broad workable ranges for all the parameters involved in your invention.**

- Detection Range: The system's detection range should encompass a wide spectrum of voice and facial expression variations, ensuring comprehensive coverage for recognizing familiar voices, identifying distress or discomfort in infants, and triggering timely alerts for appropriate intervention.

- Sensor Sensitivity: The sensitivity level of sensors must be able to detect minute traces of substances, yet avoid misleadingly false positives. Depending upon the concentration levels within a substance it should have sensing capabilities in parts per million (ppm) or even better, ppm; and parts per billion (ppb).

- Detection Time: The system should be designed to provide near real-time alerts for unrecognized voices and distress or discomfort detected through facial expressions. This requires a rapid response capability, with alerts being triggered within seconds to minutes of detecting unfamiliar voices or concerning facial expressions.

- Privacy Settings: Parameters defining privacy functions, including data encryption, storage and access control must maintain secure confidentiality. The settings should be in line with privacy laws and the organization's policy, permitting access to only authorized personnel.

- Size and Portability: The device also should be suitable in size for use in different settings, with dimensions from handheld to wall-mounted and thus portable while conveniently installable.

- Alert Mechanism: Alert. The alert system must have the ability to use either sound or sight, and should consider transmitting notification directly into selected devices or authorities at a proximity of 2-3 rooms, allowing operations in various kinds of environment.

12. References (if any):

https://link.springer.com/chapter/10.1007/978-3-031-22963-3_22

https://www.researchgate.net/publication/334897371_IOT_Based_Baby_Incubator_ for_Clinic

https://link.springer.com/chapter/10.1007/978-3-030-84245-1_3

13. Inventors Details (Full Names, Nationality and Addresses)

14. Applicant Details (Full Names, Nationality and Addresses)

15. Any additional notes or remarks.

16. For Life Sciences related inventions:

i. Provide source and geographical origin of the biological material/resource (for e.g. plants, animals, micro-organisms, their parts / genetic material andby-products with actual or potential use or value)

ii. Please note, if the biological material used in the invention is from India, then an application to seek approval of the National Biodiversity Authority(NBA) for applying for intellectual property rights (including patents) in or outside India needs to be made as per the Biological DiversityAct, 2002.

iii. Please indicate in case you need assistance to make an application to theNBA.
Yes/No

iv. Please provide sequence listing in computer readable format.

v. In case you would like us to prepare the sequence listing for submission to the Patent Office please indicate
Yes/No

vi. Please indicate if the invention relates to novel biological material for example, bacteria, fungi, eukaryotic cell lines, plant spores, genetic vectors (such as plasmids or bacteriophage vectors or viruses) containing a gene or DNA fragments, or organisms used for expression of a gene (making the protein from the DNA).
Yes/No

vii. If Yes, have you deposited material with the recognized depositary under the Budapest Treaty?

Yes/No

(Please note, in case of novel material as mentioned above, deposition mustbe made before filing of the patent application).

infant-expression

April 15, 2024

[1]:

```
!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="23zYqWKVTLknsXYVM80E")
project = rf.workspace("iot-wqguf").project("face_baby")
version = project.version(1)
dataset = version.download("folder")
```

```
Collecting roboflow
  Downloading roboflow-1.1.27-py3-none-any.whl (74 kB)
                           74.1/74.1 kB
700.0 kB/s eta 0:00:00
Collecting certifi==2023.7.22 (from roboflow)
  Downloading certifi-2023.7.22-py3-none-any.whl (158 kB)
                           158.3/158.3

kB 6.1 MB/s eta 0:00:00
Collecting chardet==4.0.0 (from roboflow)
  Downloading chardet-4.0.0-py2.py3-none-any.whl (178 kB)
                           178.7/178.7

kB 11.0 MB/s eta 0:00:00
Collecting cycler==0.10.0 (from roboflow)
  Downloading cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Collecting idna==2.10 (from roboflow)
  Downloading idna-2.10-py2.py3-none-any.whl (58 kB)
                           58.8/58.8 kB
5.3 MB/s eta 0:00:00
Requirement already satisfied: kiwisolver>=1.3.1 in
/usr/local/lib/python3.10/dist-packages (from roboflow) (1.4.5)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-
packages (from roboflow) (3.7.1)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.10/dist-
packages (from roboflow) (1.25.2)
Collecting opencv-python-headless==4.8.0.74 (from roboflow)
  Downloading opencv_python_headless-4.8.0.74-cp37-abi3-manylinux_2_17_x86_64.ma
nylinux2014_x86_64.whl (49.1 MB)
                           49.1/49.1 MB
11.9 MB/s eta 0:00:00
```

```
Requirement already satisfied: Pillow>=7.1.2 in
/usr/local/lib/python3.10/dist-packages (from roboflow) (9.4.0)
Requirement already satisfied: python-dateutil in
/usr/local/lib/python3.10/dist-packages (from roboflow) (2.8.2)
Collecting python-dotenv (from roboflow)
  Downloading python_dotenv-1.0.1-py3-none-any.whl (19 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-
packages (from roboflow) (2.31.0)
Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages
(from roboflow) (1.16.0)
Requirement already satisfied: urllib3>=1.26.6 in
/usr/local/lib/python3.10/dist-packages (from roboflow) (2.0.7)
Requirement already satisfied: tqdm>=4.41.0 in /usr/local/lib/python3.10/dist-
packages (from roboflow) (4.66.2)
Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.10/dist-
packages (from roboflow) (6.0.1)
Collecting requests-toolbelt (from roboflow)
  Downloading requests_toolbelt-1.0.0-py2.py3-none-any.whl (54 kB)
                           54.5/54.5 kB
4.9 MB/s eta 0:00:00
Collecting python-magic (from roboflow)
  Downloading python_magic-0.4.27-py2.py3-none-any.whl (13 kB)
Requirement already satisfied: contourpy>=1.0.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (1.2.1)
Requirement already satisfied: fonttools>=4.22.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (4.51.0)
Requirement already satisfied: packaging>=20.0 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (24.0)
Requirement already satisfied: pyparsing>=2.3.1 in
/usr/local/lib/python3.10/dist-packages (from matplotlib->roboflow) (3.1.2)
Requirement already satisfied: charset-normalizer<4,>=2 in
/usr/local/lib/python3.10/dist-packages (from requests->roboflow) (3.3.2)
Installing collected packages: python-magic, python-dotenv, opencv-python-
headless, idna, cycler, chardet, certifi, requests-toolbelt, roboflow
  Attempting uninstall: opencv-python-headless
    Found existing installation: opencv-python-headless 4.9.0.80
    Uninstalling opencv-python-headless-4.9.0.80:
      Successfully uninstalled opencv-python-headless-4.9.0.80
  Attempting uninstall: idna
    Found existing installation: idna 3.6
    Uninstalling idna-3.6:
      Successfully uninstalled idna-3.6
  Attempting uninstall: cycler
    Found existing installation: cycler 0.12.1
    Uninstalling cycler-0.12.1:
      Successfully uninstalled cycler-0.12.1
  Attempting uninstall: chardet
    Found existing installation: chardet 5.2.0
    Uninstalling chardet-5.2.0:
      Successfully uninstalled chardet-5.2.0
  Attempting uninstall: certifi
    Found existing installation: certifi 2024.2.2
    Uninstalling certifi-2024.2.2:
      Successfully uninstalled certifi-2024.2.2
Successfully installed certifi-2023.7.22 chardet-4.0.0 cycler-0.10.0 idna-2.10
opencv-python-headless-4.8.0.74 python-dotenv-1.0.1 python-magic-0.4.27
requests-toolbelt-1.0.0 roboflow-1.1.27

loading Roboflow workspace…
loading Roboflow project…

Downloading Dataset Version Zip in face_baby-1 to folder:: 100%|      |
3682/3682 [00:00<00:00, 13700.48it/s]



Extracting Dataset Version Zip to face_baby-1 in folder:: 100%|      | 96/96
[00:00<00:00, 2721.84it/s]
```

```
[2]: import tensorflow as tf
     from tensorflow.keras.preprocessing.image import ImageDataGenerator
     from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense

     train_dir = '/content/face_baby-1/train'
     valid_dir = '/content/face_baby-1/valid'
     test_dir = '/content/face_baby-1/test'

     image_height = 640
     image_width = 640
     batch_size = 32
     num_classes = 4

     train_datagen = ImageDataGenerator(rescale=1./255, shear_range=0.2,␣
      ↪zoom_range=0.2, horizontal_flip=True)
     valid_datagen = ImageDataGenerator(rescale=1./255)
     test_datagen = ImageDataGenerator(rescale=1./255)

     train_generator = train_datagen.flow_from_directory(train_dir,␣
      ↪target_size=(image_height, image_width), batch_size=batch_size,␣
      ↪class_mode='categorical')
     valid_generator = valid_datagen.flow_from_directory(valid_dir,␣
      ↪target_size=(image_height, image_width), batch_size=batch_size,␣
      ↪class_mode='categorical')
```

```
test_generator = test_datagen.flow_from_directory(test_dir,␣
 ↪target_size=(image_height, image_width), batch_size=batch_size,␣
 ↪class_mode='categorical', shuffle=False)

model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(image_height,␣
 ↪image_width, 3)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Flatten(),
    Dense(128, activation='relu'),
    Dense(num_classes, activation='softmax')
])

model.compile(optimizer='adam', loss='categorical_crossentropy',␣
 ↪metrics=['accuracy'])

epochs = 10
history = model.fit(train_generator, epochs=epochs,␣
 ↪validation_data=valid_generator)

test_loss, test_accuracy = model.evaluate(test_generator)
print(f'Test Loss: {test_loss}, Test Accuracy: {test_accuracy}')
```

```
Found 47 images belonging to 4 classes.
Found 11 images belonging to 1 classes.
Found 25 images belonging to 4 classes.
Epoch 1/10
2/2 [==============================] - 60s 19s/step - loss: 9.1356 - accuracy:
0.3404 - val_loss: 227.8207 - val_accuracy: 0.0000e+00
Epoch 2/10
2/2 [==============================] - 60s 41s/step - loss: 7.7498 - accuracy:
0.7234 - val_loss: 31.9492 - val_accuracy: 0.0000e+00
Epoch 3/10
2/2 [==============================] - 53s 33s/step - loss: 4.6758 - accuracy:
0.1489 - val_loss: 7.3581 - val_accuracy: 0.3636
Epoch 4/10
2/2 [==============================] - 53s 18s/step - loss: 1.2033 - accuracy:
0.4681 - val_loss: 11.6075 - val_accuracy: 0.0000e+00
Epoch 5/10
2/2 [==============================] - 52s 33s/step - loss: 1.2204 - accuracy:
0.7234 - val_loss: 11.0724 - val_accuracy: 0.0000e+00
Epoch 6/10
2/2 [==============================] - 52s 34s/step - loss: 0.8771 - accuracy:
```

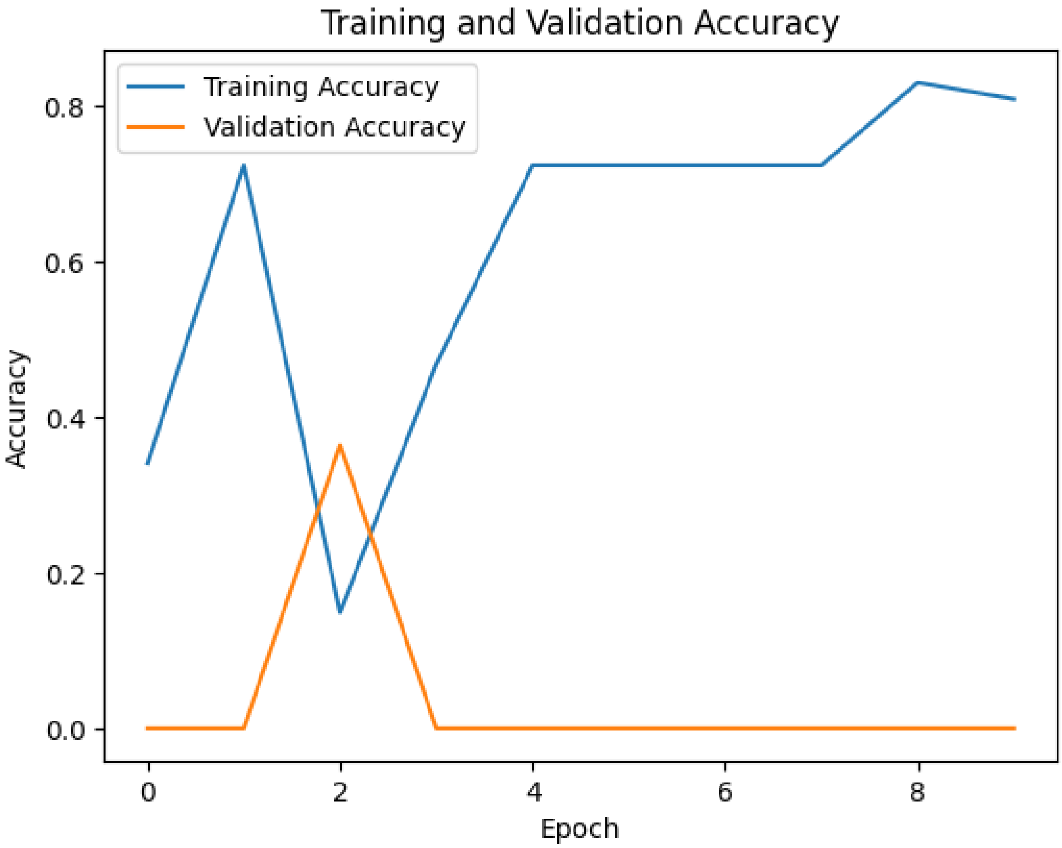
```
0.7234 - val_loss: 6.7879 - val_accuracy: 0.0000e+00
Epoch 7/10
2/2 [==============================] - 53s 20s/step - loss: 0.7674 - accuracy:
0.7234 - val_loss: 9.4336 - val_accuracy: 0.0000e+00
Epoch 8/10
2/2 [==============================] - 52s 18s/step - loss: 0.6656 - accuracy:
0.7234 - val_loss: 8.2016 - val_accuracy: 0.0000e+00
Epoch 9/10
2/2 [==============================] - 56s 18s/step - loss: 0.5209 - accuracy:
0.8298 - val_loss: 8.5406 - val_accuracy: 0.0000e+00
Epoch 10/10
2/2 [==============================] - 52s 32s/step - loss: 0.4113 - accuracy:
0.8085 - val_loss: 14.3584 - val_accuracy: 0.0000e+00
1/1 [==============================] - 8s 8s/step - loss: 0.2919 - accuracy:
0.8400
Test Loss: 0.29188480973243713, Test Accuracy: 0.8399999737739563
```

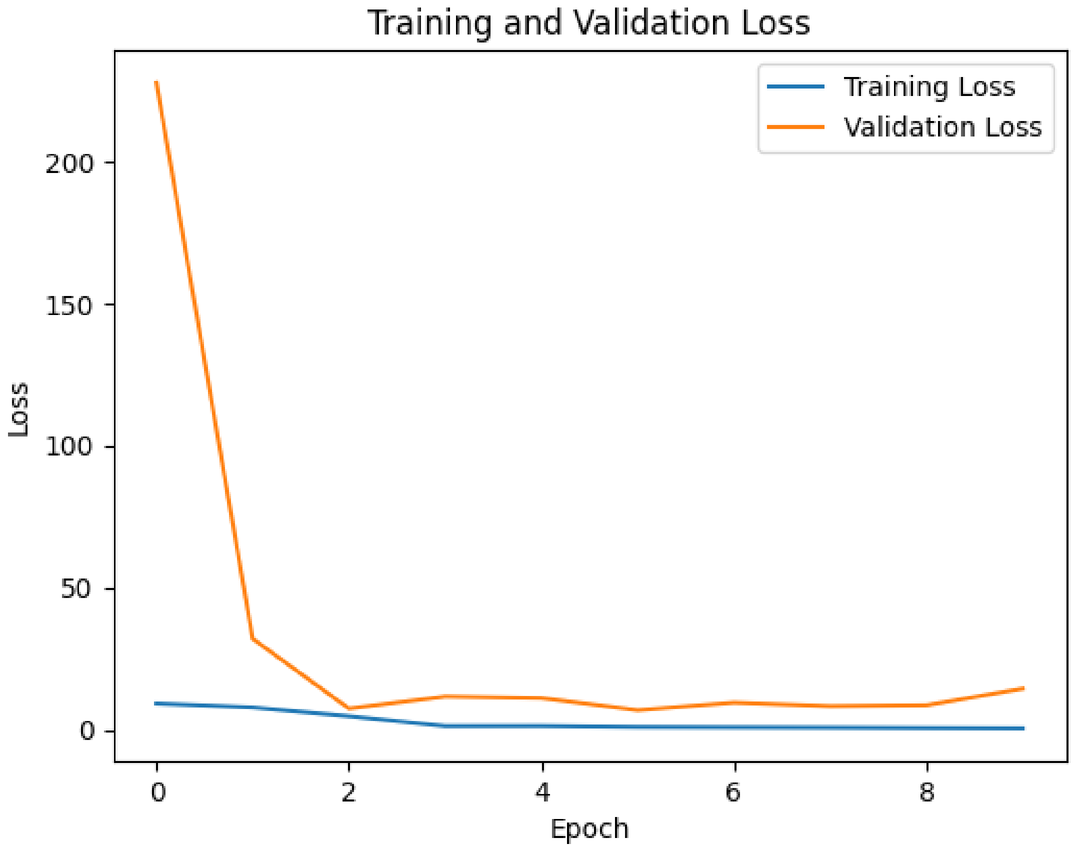
[5]:

```python
import matplotlib.pyplot as plt

plt.plot(history.history['accuracy'], label='Training Accuracy')
plt.plot(history.history['val_accuracy'], label='Validation Accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epoch')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

plt.plot(history.history['loss'], label='Training Loss')
plt.plot(history.history['val_loss'], label='Validation Loss')
plt.title('Training and Validation Loss')
plt.xlabel('Epoch')
plt.ylabel('Loss')
plt.legend()
plt.show()
```


Training and Validation Accuracy
Training Accuracy
Validation Accuracy
Accuracy
0.0
0.2
0.4
0.6
0.8
Epoch
0
2
4
6
8



[6]:

```
import numpy as np

predictions = model.predict(test_generator)

predicted_labels = np.argmax(predictions, axis=1)

true_labels = test_generator.classes

class_indices = test_generator.class_indices

class_labels = list(class_indices.keys())

true_class_names = [class_labels[i] for i in true_labels]

predicted_class_names = [class_labels[i] for i in predicted_labels]

for i in range(10):
    print(f"Image {i+1}: Predicted Class: {predicted_class_names[i]}, True␣
 ↪Class: {true_class_names[i]}")
```

```
1/1 [==============================] - 9s 9s/step
Image 1: Predicted Class: SLEEP, True Class: CRY
Image 2: Predicted Class: SLEEP, True Class: CRY
Image 3: Predicted Class: SLEEP, True Class: CRY
Image 4: Predicted Class: SLEEP, True Class: HAPPY NORMAL
Image 5: Predicted Class: NORMAL, True Class: NORMAL
Image 6: Predicted Class: SLEEP, True Class: SLEEP
Image 7: Predicted Class: SLEEP, True Class: SLEEP
Image 8: Predicted Class: SLEEP, True Class: SLEEP
Image 9: Predicted Class: SLEEP, True Class: SLEEP
Image 10: Predicted Class: SLEEP, True Class: SLEEP
```

[11]:

```python
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing import image

img_dir = '/content/download.jpg'

img = image.load_img(img_dir, target_size=(image_height, image_width))
img_array = image.img_to_array(img)
img_array = img_array / 255.0

img_array = np.expand_dims(img_array, axis=0)

predictions = model.predict(img_array)

predicted_label = np.argmax(predictions)
predicted_class = class_labels[predicted_label]

plt.imshow(img)
plt.axis('off')
plt.title(f'Predicted Class: {predicted_class}')
plt.show()
```

```
1/1 [==============================] - 0s 305ms/step
```

Predicted Class: SLEEP