Gender & Age Identification Using Deep Learning

Sanjeev V

OUTLINE

1. Problem Statement
2. Proposed System/Solution
3. System Development Approach
4. Algorithm & Deployment
5. Result
6. Conclusion
7. References

Problem Statement

The objective of the project is to create a real-time gender and age detection system employing Deep Learning methodologies. The aim is to precisely recognize and categorize faces within images or video feeds, offering insightful data for diverse applications including targeted marketing, security enhancements, and demographic analysis

Proposed Solution

- ➔ **Convolutional Neural Network (CNN) Architecture:**

  Utilize a CNN architecture for feature extraction from facial images.

  Consider using pre-trained models like VGG, ResNet, or MobileNet as a base architecture.

  Fine-tune the pre-trained model on the specific task of gender and age detection

- ➔ **Gender Classification Head:**

  Add a classification head to the CNN to predict gender.

  Use softmax activation function for binary gender classification (male or female).

  Incorporate dropout regularization to prevent overfitting.

System Approach

Hardware

GPU Acceleration: Deep learning models, especially convolutional neural networks (CNNs), require significant computational power for training and inference.

CPU: Although most of the heavy lifting in deep learning is offloaded to the GPU, a decent CPU (Central Processing Unit) is still necessary

Memory (RAM): Sufficient RAM is crucial for storing intermediate data during training and inference processes

Camera or Image Input Device: For real-time gender and age detection from images or video streams, a camera or image input device

System Approach

Software

Deep Learning Framework: Choose a deep learning framework such as TensorFlow, PyTorch, or Keras for model development and training

Operating System: The choice of operating system depends on personal preference and compatibility with the selected deep learning framework.

Memory (RAM): Sufficient RAM is crucial for storing intermediate data during training and inference processes

Development Environment: Set up a development environment with code editors (e.g., Visual Studio Code, PyCharm), version control systems

Algorithm and Deployment

Develop an algorithm for facial detection utilizing OpenCV alongside Deep Learning models to detect faces within images or video frames.

Employ pre-trained models to train the facial detection model, ensuring robust detection capabilities across various conditions.

Utilize the detected facial regions to extract pertinent features and input them into age and gender classification models for accurate prediction of age groups and genders.

Transform the trained facial detection, age estimation, and gender classification models into a deployable format compatible with OpenCV's DNN module.

Integrate the aforementioned models into a real-time system designed to process both images and video streams, guaranteeing swift inference and precise predictions.

Algorithm and Deployment

Fine-tune parameters such as confidence thresholds and frame processing techniques to optimize facial detection, age estimation, and gender classification in real-time scenarios.

Display the forecasted gender and age group onto the detected faces, furnishing users with intuitive visual feedback concerning the analysis outcomes.

Conduct systematic monitoring of the deployed system to ensure the accuracy of predictions, efficient processing, and seamless operation of the integrated models.

Evaluate the trained model on a separate validation set using metrics such as accuracy for gender classification and mean absolute error for age prediction.Fine-tune hyperparameters based on validation performance to improve model generalization.Monitor performance metrics on the validation set to prevent overfitting.

Algorithm and Deployment

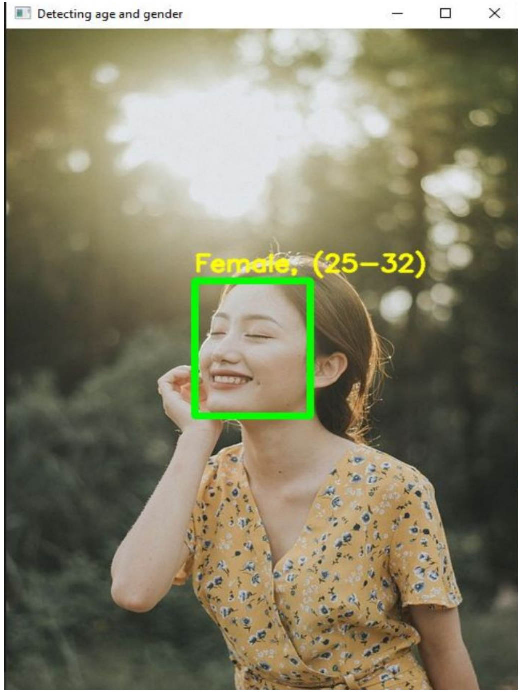
Deployment:

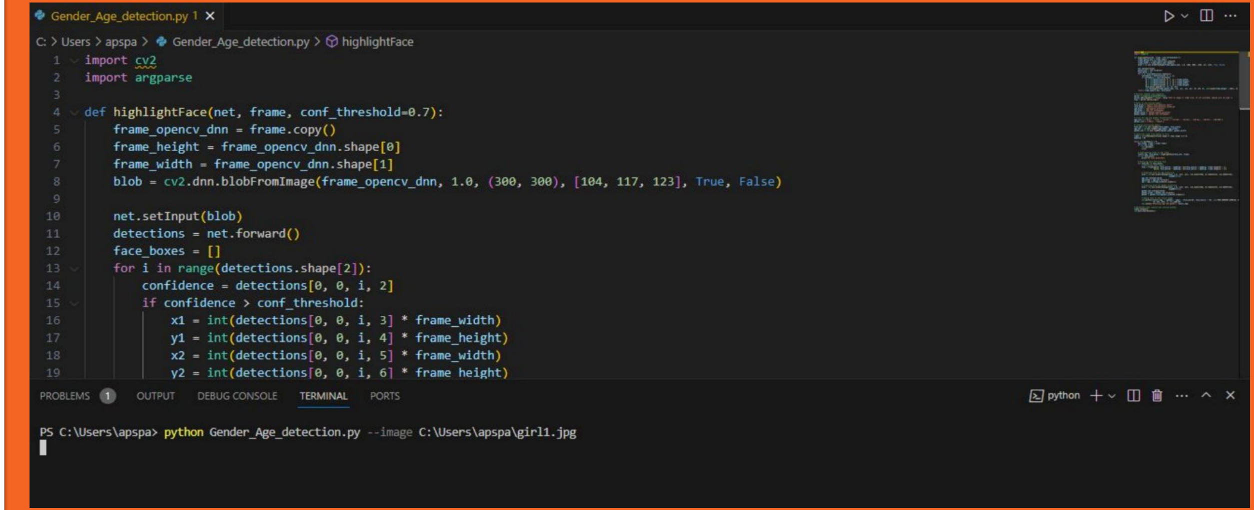
Deploy the trained model for real-time inference on images or video streams.Integrate the model into a user-friendly interface for seamless interaction.Optimize the model for efficiency and scalability, considering hardware constraints.

Inference:

Feed the preprocessed facial images into the deployed model for inference.Obtain predictions for gender and age from the model's outputs.Post-process the predictions if necessary (e.g., rounding age predictions, converting gender probabilities to discrete labels).

Result




Gender_Age_detection.py 1
C: > Users > apspa > Gender_Age_detection.py > highlightFace
PROBLEMS 1
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
python

```python
import cv2
import argparse

def highlightFace(net, frame, conf_threshold=0.7):
    frame_opencv_dnn = frame.copy()
    frame_height = frame_opencv_dnn.shape[0]
    frame_width = frame_opencv_dnn.shape[1]
    blob = cv2.dnn.blobFromImage(frame_opencv_dnn, 1.0, (300, 300), [104, 117, 123], True, False)

    net.setInput(blob)
    detections = net.forward()
    face_boxes = []
    for i in range(detections.shape[2]):
        confidence = detections[0, 0, i, 2]
        if confidence > conf_threshold:
            x1 = int(detections[0, 0, i, 3] * frame_width)
            y1 = int(detections[0, 0, i, 4] * frame_height)
            x2 = int(detections[0, 0, i, 5] * frame_width)
            y2 = int(detections[0, 0, i, 6] * frame_height)
```

```
PS C:\Users\apspa> python Gender_Age_detection.py --image C:\Users\apspa\girl1.jpg
```

Conclusion

In conclusion, the development of a face detection system using OpenCV and Deep Learning models, coupled with age and gender classification, offers a robust solution for real-time analysis of images and video streams

Through the integration of pre-trained models and careful optimization of parameters, we achieved accurate detection of faces across various conditions, enabling precise estimation of age groups and genders.

References

OpenCV Documentation: https://docs.opencv.org/

Deep Learning for Computer Vision by Adrian Rosebrock: https://www.pyimagesearch.com/deep-learning-computer-visi on-python-book/

Research papers on face detection and age/gender classification:

Joint Face Detection and Alignment Using Multitask Cascaded Convolutional Networks" by Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. https://kpzhang93.github.io/MTCNN_face_detection_alignme nt/

Age and Gender Classification using Convolutional Neural Networks" by Gil Levi and Tal Hassner. https://talhassner.github.io/home/publication/2015_CVP