MENGESAN TOPENG MUKA MENGGUNAKAN PEMBELAJARAN DALAM APLIKASI WEB

HAZIM BIN ABDUL AZIZ

DR. ZAINAL RASYID MAHAYUDDIN

Fakulti Teknologi & Sains Maklumat, Universiti Kebangsaan Malaysia, 43600 UKM Bangi, Selangor Darul Ehsan, Malaysia

ABSTRAK

Projek ini bertujuan untuk mengembangkan sistem pengesanan topeng muka yang mampu memantau dan meningkatkan pematuhan penggunaan topeng muka di tempat awam. Penggunaan topeng muka telah menjadi norma penting dalam menjaga keselamatan dan kesihatan masyarakat akibat pandemik COVID-19. Masalah utama yang ingin diselesaikan adalah kurangnya pemantauan efektif terhadap pemakaian topeng muka di tempat-tempat awam, yang dapat meningkatkan risiko penyebaran virus. Untuk menyelesaikan masalah ini, sistem pengesanan topeng muka yang menggunakan teknologi pengesanan objek canggih telah dicadangkan. Sistem ini membolehkan pengguna memuat naik gambar dan video untuk pengesanan topeng muka serta menyediakan pengesanan secara langsung melalui kamera. Strategi pembangunan melibatkan pendekatan berorientasikan objek dengan penggunaan algoritma YOLO (You Only Look Once) yang terkenal dengan kecekapan dan ketepatan dalam pengesanan objek. Reka bentuk sistem menggabungkan senibina pelayan-pelanggan untuk komunikasi yang lancar antara pengguna dan pelayan, serta penggunaan pangkalan data yang sistematik untuk penyimpanan dan pengambilan data yang cekap. Hasil akhir projek ini adalah sistem pengesanan topeng muka yang berfungsi dengan baik dan mampu memberikan keputusan yang pantas dan tepat. Sistem ini menyediakan antara muka pengguna yang mesra untuk memuat naik imej, video, dan pengesanan masa nyata, sekaligus meningkatkan efisiensi pengawasan dan pemantauan pemakaian topeng muka di tempat awam. Dengan implementasi yang luas, sistem ini diharapkan dapat membantu dalam usaha mitigasi penyebaran virus dan menjaga keselamatan masyarakat.

Kata kunci: Pengesanan Topeng Muka, COVID-19, “YOLO”

PENGENALAN

Langkah-langkah kesihatan awam baru diperlukan untuk menghalang penyebaran virus COVID-19 kerana pandemik telah mengubah norma hidup kita. Antara langkah-langkah yang harus dilakukan adalah pengambilan topeng muka yang penting. Sesetengah kajian menunjukkan bahawa memakai topeng muka boleh mengurangkan titisan pernafasan yang dilepaskan, yang bermaksud bahawa ia boleh mengurangkan penyebaran COVID-19 dan

penyakit udara yang lain. Menurut (Leung et al., 2020), Oleh kerana kerajaan dan organisasi kesihatan terus menggalakkan orang memakai topeng di kawasan awam, ia sukar untuk memastikan pematuhan, terutamanya di kawasan yang padat dengan orang dan mempunyai banyak lalu lintas.

Kaedah pengawasan manual topeng muka tidak praktikal dan memerlukan tenaga buruh, terutamanya dalam persekitaran seperti lapangan terbang, pusat membeli-belah dan pusat pengangkutan awam. Selain itu, kaedah menggunakan manusia akan sering berlakunya keletihan dan kesilapan, ini menunjukan tiada konsisten dalam protokol memakai topeng muka. Oleh itu, terdapat keperluan yang jelas untuk penyelesaian automatik yang boleh menyediakan pengesanan masa nyata dan tepat dalam penggunaan topeng muka.

Bagi menangani keperluan ini, projek ini memberi tumpuan kepada pembangunan sistem pengesanan masker wajah berasaskan web menggunakan model pengesan objek YOLOv9 (You Only Look Once). YOLOv9 terkenal kerana keseimbangan antara kelajuan dan ketepatan, menjadikannya pilihan yang ideal untuk aplikasi masa nyata (Bochkovskiy et al., 2020). Sistem ini bertujuan untuk memudahkan pemantauan yang mudah dan cekap dengan membolehkan pengguna memuat naik imej dan video untuk pengesanan topeng muka, serta membenarkan pengesan kamera langsung untuk analisis masa nyata.

Kepentingan projek ini ditegaskan oleh keperluan berterusan untuk mengekalkan langkah-langkah keselamatan kesihatan awam dalam menghadapi penyakit berjangkit yang berkembang. Sistem pengesanan wajah automatik boleh memainkan peranan penting dalam menyokong langkah-langkah ini dengan menyediakan penyelesaian pemantauan yang konsisten, boleh diperluas dan tidak mengganggu. Dengan memanfaatkan teknik pembelajaran mesin canggih dan teknologi web yang boleh diakses, projek ini bertujuan untuk menyumbang kepada usaha yang lebih luas untuk melindungi kesihatan awam sambil menyesuaikan diri dengan norma baru yang diperlukan oleh pandemi.

METODOLOGI KAJIAN

Dalam bab ini, saya akan menerangkan proses pembangunan sistem pengesanan topeng muka menggunakan teknologi YOLOv9. Bab ini akan merangkumi langkah-langkah utama dalam pembangunan sistem ini, termasuk pengumpulan dan pra-pemprosesan data, latihan model, penilaian model dan integrasi aplikasi web.

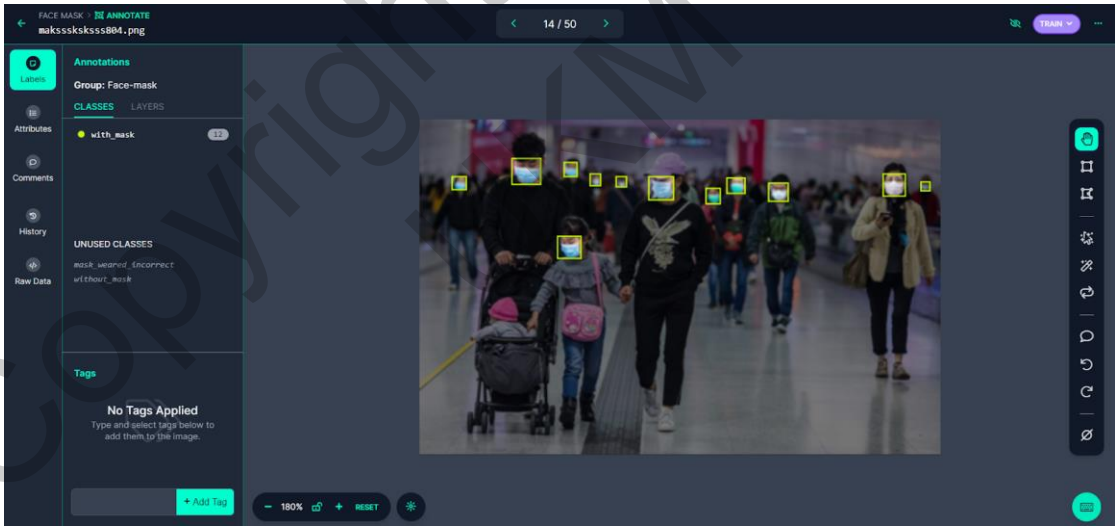
YOLOv9 merupakan sistem pengesanan objek yang canggih yang memboleh mengenal pasti objek dalam masa nyata dengan ketepatan yang tinggi. Keupayaan untuk mengesan objek secara langsung dalam satu pemprosesan adalah sangat sesuai untuk aplikasi pengesanan topeng muka. Kelebihan utama YOLOv9 termasuk kebolehan untuk mengendalikan pemprosesan secara langsung, membolehkan sistem untuk mengesan topeng muka dalam waktu nyata tanpa perlu menunggu sistem mengesan muka dalam tempoh yang lama. Selain itu, pilihan untuk menggunakan teknologi YOLOv9 untuk sistem pengesan topeng muka disebabkan ketepatan yang tinggi.

Fasa Pengumpulan Data Dan Pra-pemprosesan

Untuk pengumpulan data, set data yang tersedia di laman sesawang telah digunakan untuk melatih model pengesanan topeng muka menggunakan YOLOv9. Set data ini terdiri daripada gambar-gambar individu yang memakai topeng muka, tidak memakai topeng muka, dan memakai topeng muka dengan cara yang salah. Setiap gambar dalam set data ini perlu dianotasi, ini merupakan langkah penting dalam proses pembelajaran terkawal untuk melatih model YOLOv9.

Penggunaan Roboflow membantu dalam menganotasikan set data seperti yang ditunjukkan dalam Rajah 4.1. Anotasi ini memastikan setiap objek dalam gambar dikenal pasti dan diberi label yang sesuai untuk melatih model. Sebanyak 586 gambar telah digunakan dalam latihan, 168 gambar untuk pengesahan, 84 gambar untuk ujian, menjadikan jumlah pengumpulan data sebanyak 838 gambar.

Selain itu, beberapa langkah pra-pemprosesan data telah dijalankan untuk menyediakan set data untuk latihan. Langkah-langkah ini termasuk menyesuaikan automatik orientasi imej, penyesuaian saiz setiap imej kepada 640x640 piksel, dan melaksanakan teknik flip dan potong untuk meningkatkan kepelbagaian set data. Langkah-langkah pra-pemprosesan ini penting untuk memastikan kualiti dan kecekapan latihan model YOLOv9.



Rajah 4.1Anotasi Menggunakan Robotflow

Fasa Latihan Model

Dalam proses latihan model, pustaka Ultralytics seperti yang ditunjukkan dalam Rajah 4.2 telah digunakan. Pustaka ini mengandungi keperluan seperti YOLO untuk melatih model pengesanan topeng muka dan memudahkan proses latihan serta penyesuaian parameter bagi memperoleh prestasi yang optimum dalam pengesanan objek.

```
!pip install ultralytics
```

Rajah 4.2 Muat Turun Pustaka Ultralytics

Roboflow menyediakan API untuk memudahkan akses kepada set data yang telah dianotasi olehnya. Rajah 4.3 menunjukkan cara pustaka Roboflow digunakan untuk memuat turun data yang dianotasi daripada projek Roboflow.

```
from roboflow import Roboflow
rf = Roboflow(api_key="I2bdOyeLbLrh8ztNrbzQ")
project = rf.workspace("robot-aplikasi").project("face-mask-qw99q")
version = project.version(3)
dataset = version.download("yolov9")
```

Rajah 4.3 Mengeluarkan Set Data Daripada Robotflow

Dalam kod di atas, pustaka Roboflow diimport dan objek Roboflow dianalisis dengan kunci API yang diperolehi dari laman web Roboflow. Projek yang ingin diakses dalam ruang kerja Roboflow dan versi set data yang ingin dimuat turun telah ditetapkan. Akhirnya, set data yang dianotasi dalam format YOLOv9 dimuat turun menggunakan fungsi "download()".

Latihan model YOLOv9 dilakukan dengan menggunakan data yang dikumpulkan dan pra-pemprosesan sebelumnya. Latihan model telah dilaksanakan dengan 25 epochs dan kadar pembelajaran sebanyak 0.001. Set data yang telah dikumpulkan, yang terdiri daripada gambar orang dengan topeng muka yang dipakai dengan betul, topeng muka yang dipakai dengan salah, dan tiada topeng muka, telah digunakan untuk melatih model dari awal. Tujuan proses ini adalah untuk membolehkan model mempelajari ciri visual yang berkaitan dengan setiap kategori objek dan meningkatkan keupayaannya untuk membezakan topeng muka dengan tepat dalam pelbagai keadaan.

Rajah 4.4 menunjukkan kod yang digunakan untuk melatih model menggunakan YOLO dengan set data yang telah tersedia. Parameter "task=detect" menunjukkan bahawa tugas pengesanan objek sedang dilaksanakan, "mode=train" menunjukkan bahawa model sedang dilatih, "model=yolov9s.pt" menunjukkan model YOLOv9 yang digunakan, "data={dataset.location}/data.yaml" menunjukkan lokasi set data dan fail konfigurasi data YAML yang menyimpan maklumat mengenai set data yang digunakan, "epochs=25" menunjukkan jumlah epochs latihan, dan "imgsz=640" menetapkan saiz imej input yang digunakan semasa latihan model.

```
!yolo task=detect mode=train model=yolov8s.pt data={dataset.location}/data.yaml epochs=25 imgsz=640
                   all        168        846      0.836      0.695      0.761      0.485

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      18/25      4.17G      1.102     0.6214      1.056         28        640: 100% 37/37 [00:11<00:00,  3.25it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:02<00:00,  2.06it/s]
                   all        168        846      0.804      0.678      0.745      0.477

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      19/25      4.16G      1.074     0.6052      1.048        119        640: 100% 37/37 [00:11<00:00,  3.21it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:02<00:00,  2.59it/s]
                   all        168        846      0.847      0.731      0.795      0.509

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      20/25      4.33G      1.046     0.5797      1.047         41        640: 100% 37/37 [00:11<00:00,  3.09it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:01<00:00,  3.09it/s]
                   all        168        846      0.827      0.722      0.783      0.508

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      21/25      4.38G      1.048     0.5639      1.033        108        640: 100% 37/37 [00:12<00:00,  3.05it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:01<00:00,  3.01it/s]
                   all        168        846      0.828      0.746      0.783      0.504

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      22/25      4.32G       1.03     0.5451      1.019         55        640: 100% 37/37 [00:12<00:00,  3.08it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:02<00:00,  2.98it/s]
                   all        168        846      0.851      0.713      0.791      0.509

      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      23/25       4.4G       1.01     0.5275      1.021         38        640: 100% 37/37 [00:11<00:00,  3.12it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:02<00:00,  2.92it/s]
                   all        168        846      0.848      0.729      0.784      0.507
```

Rajah 4.4 Kod Melatih Model

Fasa Penilaian Model

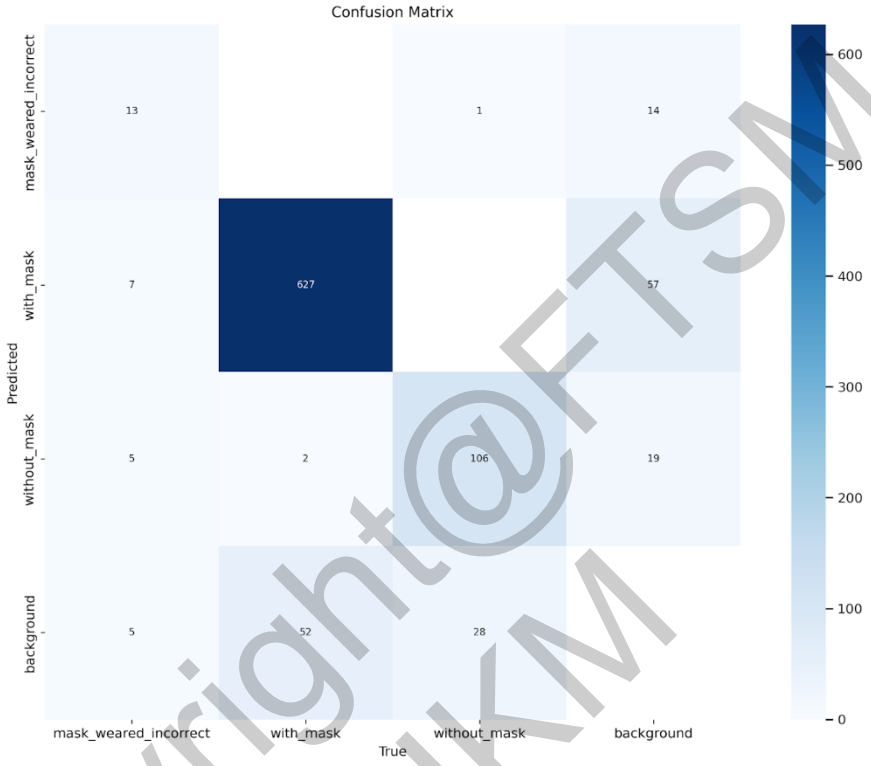
Semua langkah yang diperlukan untuk menilai prestasi model YOLOv9 yang telah dilatih telah dijalankan dengan teliti. Ketepatan (precision), pengingatan (recall), dan kesepakatan purata (mAP) adalah metrik utama yang digunakan untuk penilaian. Pengingatan mengukur keupayaan model untuk mengenal pasti semua objek sebenar sebagai objek yang dijangka, manakala ketepatan merujuk kepada kebolehan model untuk mengenali objek sebenar sebagai objek yang dijangka. Nilai ketepatan dan pengingatan untuk setiap kelas objek ditakrifkan sebagai perjanjian purata (mAP).

```
Validating runs/detect/train5/weights/best.pt...
Ultralytics YOLOv8.2.6 🚀 Python-3.10.12 torch-2.2.1+cu121 CUDA:0 (Tesla T4, 15102MiB)
Model summary (fused): 168 layers, 11126745 parameters, 0 gradients, 28.4 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 6/6 [00:06<00:00,  1.11s/it]
                   all        168        846      0.875      0.739      0.815      0.529
 mask_weared_incorrect        168         30       0.79        0.6       0.68      0.433
             with_mask        168        681      0.944      0.891       0.95      0.657
          without_mask        168        135      0.891      0.726      0.814      0.497
Speed: 0.5ms preprocess, 5.3ms inference, 0.0ms loss, 5.4ms postprocess per image
```

Rajah 4.5 Hasil Penilaian Model

Rajah 4.5 menunjukkan bahawa nilai ketepatan (Precision) adalah 0.875 untuk kelas "semua," nilai pengingatan (Recall) adalah 0.739, dan kesepakatan purata (mAP50) adalah 0.815, yang menunjukkan kemampuan model untuk membezakan pelbagai jenis topeng muka. Bagi kelas "mask_weared_incorrect," model mendapat nilai ketepatan 0.79, pengingatan 0.6, dan kesepakatan purata (mAP50) adalah 0.68, menunjukkan prestasi yang sedikit lebih rendah berbanding kelas "semua." Dengan nilai 0.944, pengingatan 0.891, dan kesepakatan purata (mAP50) yang sangat tinggi pada 0.95, model menunjukkan prestasi cemerlang dalam mengenal pasti orang yang memakai topeng muka dengan betul bagi kelas "with_mask." Bagi kelas "without_mask," model mencapai ketepatan 0.891, pengingatan 0.726, dan kesepakatan purata (mAP50) sebanyak 0.814, menunjukkan prestasi yang baik dalam mengenal pasti individu yang tidak memakai topeng muka.

Selain daripada menilai model menggunakan metrik seperti ketepatan (precision), pengingatan (recall), dan kesepakatan purata (mAP50), matriks kekeliruan juga merupakan alat penting untuk menilai prestasi model pengesanan. Matriks kekeliruan memaparkan jumlah prediksi yang tepat dan tidak tepat yang dibuat oleh model untuk setiap kelas objek. Ini membantu kita memahami sejauh mana model dapat mengenal pasti objek dengan tepat.



Rajah 4.6 Matriks Kekalutan Peta Haba

Rajah 4.6 menunjukkan peta haba yang digunakan untuk memperlihatkan matriks kekeliruan dengan warna-warna yang berbeza, memperlihatkan kepadatan data dalam setiap sel. Warna-warna yang berbeza membantu melihat pola dengan lebih jelas, memudahkan interpretasi hasil. Dalam peta haba, warna yang lebih terang mewakili jumlah yang tinggi manakala warna yang malap mewakili jumlah yang rendah. Kesimpulannya, jumlah positif sebenar yang tinggi menunjukkan kelas itu dapat mengesan objek dengan cemerlang manakala jumlah yang rendah menunjukkan model menghadapi cabaran dalam mengenal pasti objek.

Fasa Integrasi Aplikasi Web

Untuk mengintegrasikan model YOLOv9 yang telah dilatih ke dalam aplikasi web yang menggunakan pengesanan topeng muka masa nyata, kerangka kerja Flask, Visual Studio Code (VSCode), dan persekitaran PyTorch telah digunakan. Dalam pembangunan aplikasi web, Flask berfungsi sebagai rangka kerja utama. VSCode menawarkan kemudahan yang diperlukan untuk integrasi model YOLOv9 ke dalam aplikasi web dengan mudah. Pemprosesan model PyTorch adalah penting kerana ia menyediakan keupayaan yang perlu

untuk menyambungkan model yang telah dilatih ke aplikasi web.

Selain itu, VSCode memudahkan proses penulisan, pengujian, dan penyebaran aplikasi web dengan ciri-ciri pembangunan yang lengkap, termasuk integrasi model YOLOv9. Terdapat beberapa perkara yang perlu dipertimbangkan untuk mengoptimumkan model untuk penyebaran dalam persekitaran web. Menjamin kelajuan inferensi model yang ideal dan penggunaan sumber yang berkesan dalam persatuan web adalah dua daripada ini. Ini dilakukan untuk memastikan aplikasi web boleh berfungsi dengan lancar dan responsif untuk mengesan topeng muka dalam masa nyata.

```
if __name__ == "__main__":
    parser = argparse.ArgumentParser(description="Flask app exposing yolov8 models")
    parser.add_argument("--port", default=5000, type=int, help="port number")
    args = parser.parse_args()
    model = YOLO(r'C:\Users\Hazim\Documents\FYP Demo\demo4\facemaskv8.pt')
    app.run(host="0.0.0.0", port=args.port)
```

Rajah 4.7 Kod Memulaan Flask App

Rajah 4.7 menunjukkan contoh pelaksanaan Flask app sebagai skrip utama. Argumen "--port" ditambahkan untuk menentukan nombor port Flask app. Nilai lalai untuk port ditetapkan kepada 5000 jika tiada nilai diberikan. Kemudian, kami menggunakan parse_args() untuk menguraikan argumen yang diberikan semasa pelaksanaan skrip dan menyimpannya dalam objek args. Objek ‘model’ telah diberikan laluan kepada facemaskv9.pt, di mana berat telah diperolehi daripada latihan model sebelum ini.
Kod ini menjalankan Flask app pada alamat IP yang dinyatakan oleh "0.0.0.0" dan nombor port yang ditentukan oleh argumen "--port" yang diberikan semasa pelaksanaan skrip. Apabila kod ini dijalankan sebagai skrip utama, ia akan memulakan Flask app yang mengekspos model YOLOv9 dan boleh menerima permintaan HTTP untuk pengesanan objek

```
@app.route("/", methods=["GET", "POST"])
def predict_img():
    if request.method == "POST":
        if 'file' in request.files:
            f = request.files['file']
            basepath = os.path.dirname(__file__)
            filepath = os.path.join(basepath,'uploads',f.filename)
            print("upload folder is ", filepath)
            f.save(filepath)
            global imgpath
            predict_img.imgpath = f.filename
            print("printing predict_img :::::: ", predict_img)

            file_extension = f.filename.rsplit('.', 1)[1].lower()

            if file_extension.lower() in ['jpg', 'jpeg', 'png']:
                img = cv2.imread(filepath)

                # Perform the detection
                model = YOLO(r'C:\Users\Hazim\Documents\FYP Demo\demo4\facemaskv8.pt')
                detections = model(img, save=True)
                return display(f.filename)
```

Rajah 4.8 Fungsi MeramalImej

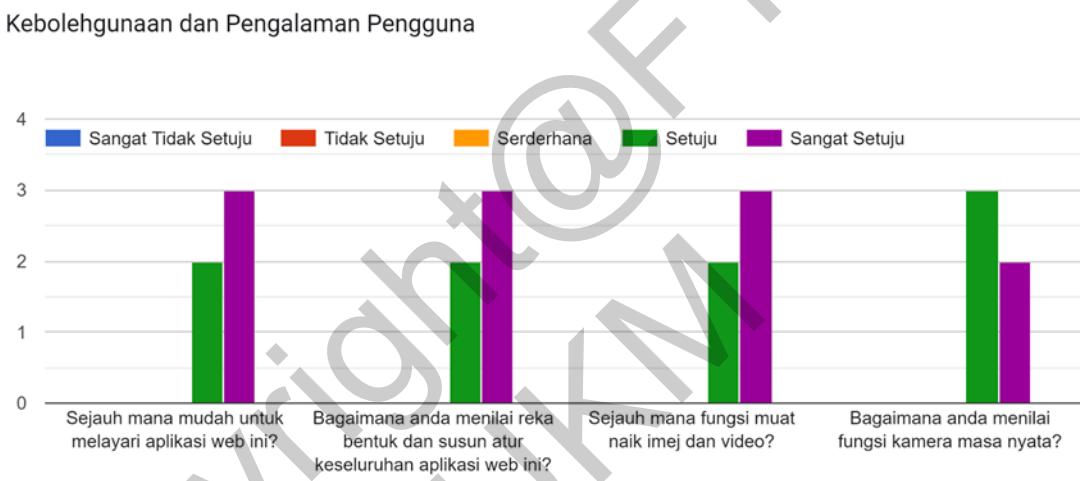
Dalam rajah 4.8 menunjukkan kod fungsi meramal imej. Di sini, "@app.route("/", methods=["GET", "POST"])" menentukan laluan untuk URL utama ("/") dan membenarkan dua kaedah HTTP: GET dan POST. Seterusnya untuk kaedah pemeriksaan permintaan, “if request.method == "POST":” memeriksa jika kaedah permintaan adalah POST, yang menunjukkan bahawa data (imej gagal) sedang dihantar ke pelayan. Fungis kod ini “f = request.files['file']” mengambil fail yang dihantar. Kemudian menyimpan nama fail dalam variabel global menggunakan kod ini “predict_img.imgpath = f.filename”. Untuk memastikan fail yang dimuat naik adalah imej dengan format yang diterima seperti jpg, jpeg dan png, kod ini melakukannya “if file_extension.lower() in ['jpg', 'jpeg', 'png']:”. Di bawah ini ialah bahagian penting di mana analisis imej dilakukan. img = cv2.imread(filepath) memuat naik gambar menggunakan “OpenCV” . Seterusnya , “model = YOLO (r'C:\Users\Hazim\Documents\FYP Demo\demo4\facemaskv9.pt')” memuat turun model YOLO yang telah dilatih sebelumnya. Selain itu, “detections = model(img, save=True)” melakukan pengesanan objek pada gambar yang dimuat naik dan menyimpan hasilnya “return display (f.filename)” mengembalikan nama fail untuk dipaparkan.

```
elif file_extension == 'mp4':
    video_path = filepath  # replace with your video path
    cap = cv2.VideoCapture(video_path)

    # get video dimensions
    frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))

    # Define the codec and create VideoWriter object
    fourcc = cv2.VideoWriter_fourcc(*'mp4v')
    out = cv2.VideoWriter('output.mp4', fourcc, 30.0, (frame_width, frame_height))

    # initialize the YOLOv8 model here
    model = YOLO(r'C:\Users\Hazim\Documents\FYP Demo\demo4\facemaskv8.pt')

    while cap.isOpened():
        ret, frame = cap.read()
        if not ret:
            break

        # do YOLOv9 detection on the frame here
        #model = YOLO('yolov9c.pt')
        results = model(frame, save=True)  #working
        print(results)
        cv2.waitKey(1)

        res_plotted = results[0].plot()
        cv2.imshow("result", res_plotted)

        # write the frame to the output video
        out.write(res_plotted)

        if cv2.waitKey(1) == ord('q'):
            break

    return video_feed()
```

Rajah 4.9 Fungsi Meramal Video

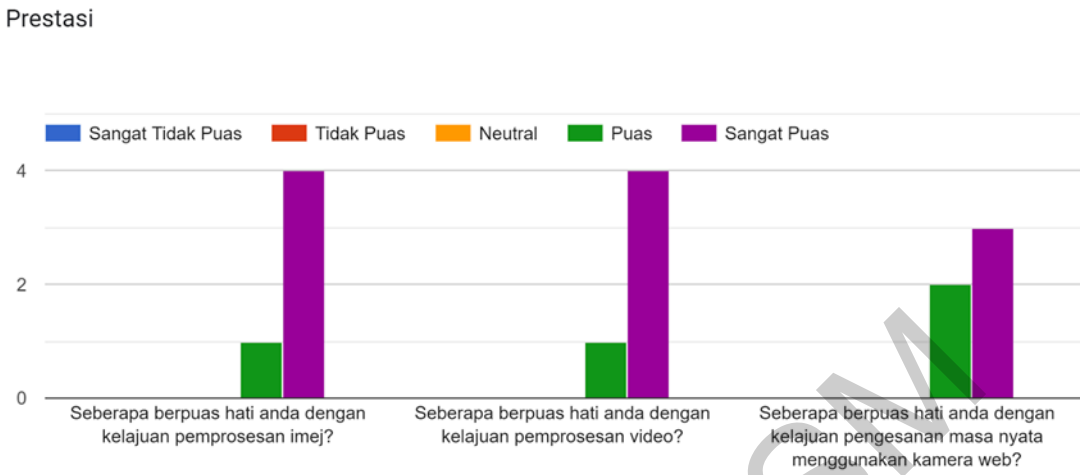
Di dalam Rajah 4.9 mengenai kod yang mengawal pemprosesan video. Fungsi ini bermula dengan memastikan bahawa fail yang diterima adalah video dalam format "mp4" dan memulakan objek "videoCapture" untuk membaca video itu. Kemudian, dapatkan dimensi video untuk digunakan semasa menulis hasil video. Kemudian, objek “VideoWriter” dibuat untuk menyimpan video yang telah diproses dengan pengesan YOLOv9. Kod “cap.read()” ialah untuk membaca bingkai video dan kemiringan binkai tersebut akan menjalankan pengesanan YOLOV9. Untuk visualisasi dan penyimpanan, “res_plotted = results[0].plot()” plot hasil pengesan, “cv2.imshow("result", res_ploted)” memaparkan hasil pada skrin, dan “out.write(res_protted)” menulis bingkai yang telah dikendalikan ke output video. Selepas selesai, untuk membolehkan pengguna menghentikan pemprosesan dengan menekan 'q'.

KEPUTUSAN PENGUJIAN DAN PERBINCANGAN



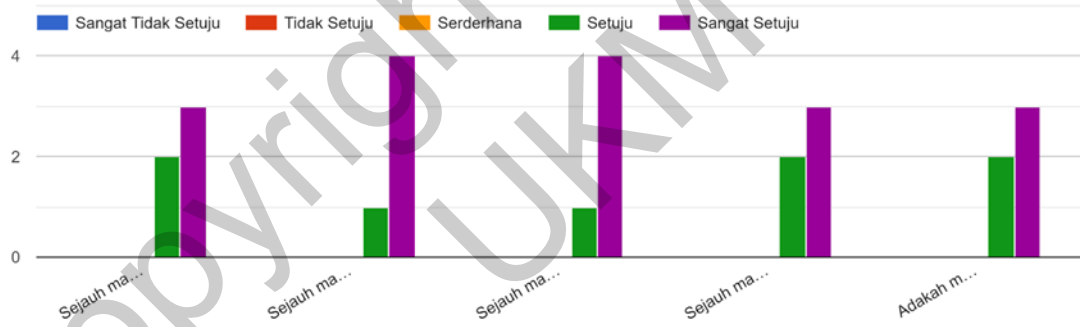
Rajah 4.16 Keputusan Kebolehgunaan dan Pengalaman Pengguna

Keputusan kebolehgunaan dan pengalaman pengguna adalah ditunjukkan dalam rajah 4.16 Para pengguna memberikan maklum balas yang positif mengenai kebolehgunaan aplikasi web. Soalan-soalan mengenai kemudahan melayari aplikasi, reka bentuk dan susun atur, serta fungsi muat naik imej dan video mendapat respons 2 "Setuju" dan 3 "Sangat Setuju". Fungsi kamera masa nyata juga dinilai positif dengan 3 pengguna "Setuju" dan 2 pengguna "Sangat Setuju".



Rajah 4.17 Keputusan Prestasi

Pada rajah 4.17, bahagian prestasi semua pengguna sangat berpuas hati dengan kelajuan pemprosesan imej dan video, dengan 1 pengguna menyatakan "Puas" dan 4 pengguna "Sangat Puas". Kelajuan pengesanan masa nyata juga dinilai tinggi, dengan 2 pengguna "Puas" dan 3 pengguna "Sangat Puas".



Rajah 4.18 Keputusan Prestasi model YOLOv9

Dalam Rajah 4.18 prestasi model YOLOv9 menunjukkan prestasi yang baik dalam pengesanan topeng muka. Pengesanan untuk imej yang dimuat naik mendapat 2 "Setuju" dan 3 "Sangat Setuju". Pengesanan untuk video dan pengesanan masa nyata mendapat respons 1 "Setuju" dan 4 "Sangat Setuju". Ketepatan dan kejelasan maklum balas visual juga dinilai tinggi dengan 2 "Setuju" dan 3 "Sangat Setuju".

Rajah 4.19 Keputusan Kesuluran Pengguna

Keputusan keseluruhan pengguna umumnya berpuas hati dengan prestasi aplikasi web seperti yang ditunjukkan oleh Rajah 4.19, dengan 80% sangat berkemungkinan untuk berpuas hati dan 20% berkemungkinan berpuas hati. Beberapa pengguna melaporkan masalah teknikal seperti tempoh muat naik yang agak lama dan aplikasi terhenti. Kebanyakan pengguna sangat mungkin akan mengesyorkan aplikasi ini kepada orang lain.

Antara ciri yang paling dihargai termasuk pengesanan kamera masa nyata, perlindungan bakteria, dan ketepatan pengesanan topeng muka untuk video yang dimuat naik. Bagi ciri-ciri kurang berguna atau boleh diperbaiki seperti beberapa pengguna mencadangkan penambahbaikan pada tempoh muat naik pengesanan video. Bagi komen atau cadangan tambahan. Secara keseluruhan, maklum balas tambahan sangat positif, dengan beberapa pengguna menyatakan bahawa aplikasi ini mampu mengesan dan membezakan jenis topeng muka dengan tepat semasa memasuki kawasan jangkitan COVID-19. Dengan hasil pengujian ini, kita dapat memahami kekuatan dan kelemahan aplikasi web dan menghasilkan penambahbaikan yang diperlukan berdasarkan maklum balas pengguna.

Cadangan Penambahbaikan

Selepas menjalankan kajian yang menyeluruh, beberapa cadangan penambahbaikan termasuk meningkatkan kapasiti pelayan, menambah dokumentasi terperinci bagi setiap komponen sistem, dan menambah fungsi tambahan seperti analisis statistik penggunaan topeng muka dan pemberitahuan kepada pihak berkuasa tempatan. Selain itu, meneroka penggunaan model-model pengesanan objek lain yang lebih canggih atau menggabungkan beberapa model dapat meningkatkan ketepatan dan keberkesanan sistem.

KESIMPULAN

Projek pengesanan topeng muka menggunakan YOLOv9 dalam aplikasi web berasaskan Flask telah mencapai banyak pencapaian signifikan, meskipun menghadapi beberapa kekangan. Kekuatan utama sistem ini terletak pada ketepatan yang tinggi dalam pengesanan topeng muka, di mana model YOLOv9 menunjukkan skor ketepatan, pengingat, dan mAP

yang sangat baik. Integrasi dengan aplikasi web membolehkan pengesanan masa nyata, memberikan hasil yang pantas dan tepat kepada pengguna. Di samping itu, antara muka pengguna yang mesra pengguna memudahkan pemuatan gambar dan video, serta memaparkan hasil pengesanan dengan jelas dan cepat. Penggunaan teknologi terkini seperti YOLOv9 dan Flask memperlihatkan keupayaan untuk mengaplikasikan teknologi terbaru dalam bidang pengesanan objek dan pengembangan aplikasi web.

Walau bagaimanapun, beberapa kekangan telah dihadapi sepanjang projek ini. Kekurangan dokumentasi terperinci untuk YOLOv9 menyebabkan beberapa masalah teknikal yang memerlukan masa tambahan untuk diselesaikan. Di samping itu, kekurangan alat bantu yang sesuai untuk anotasi data memerlukan penggunaan alat pihak ketiga seperti Roboflow, yang menambah kompleksiti dalam proses pembangunan. Isu prestasi seperti tempoh muat naik yang agak lama dan aplikasi yang terhenti menunjukkan keperluan untuk penambahbaikan lanjut bagi meningkatkan prestasi sistem. Kapasiti pelayan juga perlu ditingkatkan untuk menangani beban kerja yang lebih tinggi dan memastikan kestabilan sistem dalam keadaan penggunaan yang intensif.

Bagi penambahbaikan di masa hadapan, adalah penting untuk menyediakan dokumentasi yang lebih terperinci bagi setiap komponen sistem, khususnya untuk model YOLOv9, untuk memudahkan proses pembangunan dan penyelenggaraan. Kod yang optimum dan peningkatan kapasiti pelayan perlu dilakukan secara berterusan untuk mengurangkan tempoh muat naik dan mengelakkan aplikasi daripada terhenti. Selain itu, menambah fungsi tambahan seperti analisis statistik penggunaan topeng muka, laporan berkala, dan pemberitahuan untuk pihak berkuasa tempatan dapat meningkatkan utiliti sistem. Maklum balas pengguna juga perlu diambil kira untuk penambahbaikan berterusan, termasuk memperbaiki antara muka pengguna dan menambah ciri-ciri baru yang berguna.

Akhir sekali, meneroka penggunaan model-model pengesanan objek lain yang lebih canggih atau menggabungkan beberapa model dapat meningkatkan ketepatan dan keberkesanan sistem.
Secara keseluruhannya, projek ini berjaya mencapai objektif utamanya untuk membangunkan sistem pengesanan topeng muka yang tepat dan boleh dipercayai. Walaupun terdapat beberapa kekangan yang dihadapi, langkah-langkah penambahbaikan yang dicadangkan akan memastikan sistem ini terus berkembang dan memberikan nilai yang lebih besar dalam usaha memantau dan meningkatkan pematuhan penggunaan topeng muka di tempat awam..

PENGHARGAAN

Dengan menyebut nama Allah yang Maha Pemurah lagi Maha Penyayang, segala puji bagi Allah. Alhamdulillah, dengan izin dan limpahan rezeki dari Allah S.W.T, saya berjaya menyiapkan tesis ini.

Saya ingin mengucapkan terima kasih kepada penyelia saya, Dr. Zainal Rasyid Mahayuddin, yang saya hormati. Beliau telah memberikan panduan dan bimbingan yang berharga, serta

membantu saya sepenuhnya dalam menyelesaikan tesis ini. Juga, terima kasih kepada rakan-rakan seperjuangan saya yang sentiasa memberikan sokongan dan semangat untuk saya menyelesaikan karya ini.

Akhir sekali, setinggi-tinggi terima kasih kepada ibu saya, Hasnah Binti Omar, yang sentiasa mendoakan kejayaan anaknya. Tanpa redha dan doanya, saya tidak mungkin mencapai tahap ini. Juga, terima kasih kepada ayah saya, Abdul Aziz Bin Mahmud, yang memberikan nasihat yang berharga

RUJUKAN

Acharya, A. 2024. YOLOv9: SOTA Object Detection Model Explained. https://encord.com/blog/yolov9-sota-machine-learning-object-dection-model/.

Asana, T. 2024. Flowchart 101: Symbols, Types, and How to Create Them [2024] • Asana. Asana.

Bochkovskiy, A., Wang, C.-Y. & Liao, H.-Y.M. 2020. YOLOv4: Optimal Speed and Accuracy of Object Detection. ResearchGate.

DeepSource | The Modern Static Analysis Platform. (n.d.). . https://deepsource.com/glossary/incremental-development#:~:text=Incremental%20development%20is%20a%20method,knows%20the%20end%20result%20clearly.

Dhapare, P., Agarwal, A. & Doshi, D. 2019. Detection of Counterfeit Currency using Image Processing Techniques. 2019 IEEE 5th International Conference for Convergence in Technology (I2CT).

Easa, H.K., Saber, A.A., Hamid, N.K. & Saber, H.A. 2023. Machine learning based approach for detection of fake banknotes using support vector machine. Indonesian Journal of Electrical Engineering and Computer Science 31(2): 1016.

Face Masks and COVID-19. 2024. . https://newsinhealth.nih.gov/2021/11/face-masks-covid-19.

Flowchart - Process Flow Charts, Templates, How To, and More. (n.d.). . https://www.smartdraw.com/flowchart/.

GeeksforGeeks. 2020. Difference between Structure chart and Flow chart. https://www.geeksforgeeks.org/difference-between-structure-chart-and-flow-chart/.

GeeksforGeeks. 2022. Object Detection vs Object Recognition vs Image Segmentation. https://www.geeksforgeeks.org/object-detection-vs-object-recognition-vs-image-segmentation/.

GeeksforGeeks. 2023. Database Design in DBMS. https://www.geeksforgeeks.org/database-design-in-dbms/.

GeeksforGeeks. 2023. Software Requirement Specification (SRS) Format. https://www.geeksforgeeks.org/software-requirement-specification-srs-format/.

GeeksforGeeks. 2024. Black Box Testing Software Engineering. https://www.geeksforgeeks.org/software-engineering-black-box-testing/.

GeeksforGeeks. 2024. Sequence Diagrams Unified Modeling Language (UML). https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/.

GeeksforGeeks. 2024a. Algorithms Design Techniques. https://www.geeksforgeeks.org/algorithms-design-techniques/.

GeeksforGeeks. 2024b. Convolutional Neural Network (CNN) in Machine Learning. https://www.geeksforgeeks.org/convolutional-neural-network-cnn-in-machine-learning/.

GeeksforGeeks. 2024c. Client-Server Model. https://www.geeksforgeeks.org/client-server-model/.

GeeksforGeeks. 2024d. User Interface Design. https://www.geeksforgeeks.org/software-engineering-user-interface-design/.

Gibbons, S. 2024. User Need Statements: The 'Define' Stage in Design Thinking. https://www.nngroup.com/articles/user-need-statements/.

How well do face masks protect against COVID-19? 2023. . https://www.mayoclinic.org/diseases-conditions/coronavirus/in-depth/coronavirus-mask/art-20485449.

Ibrahim, M.A. 2023. Automatically designing CNNs architectures using genetic algorithm. Medium.

Kamble, K., Bhansali, A., Satalgaonkar, P. & Alagundgi, S. 2019. Counterfeit Currency Detection using Deep Convolutional Neural Network. 2019 IEEE Pune Section International Conference (PuneCon).

Lane, G.K.C. 2023. How to Write an SRS Document (Software Requirements Specification Document). https://www.perforce.com/blog/alm/how-write-software-requirements-specification-srs-document#:~:text=A%20system%20requirements%20specification%20(abbreviated,an%20analysis%20of%20business%20needs.

Lee, S., & Lee, H. (2018). Counterfeit Bill Detection Algorithm using Deep Learning.

Leung, K., Wu, J.T., Liu, D. & Leung, G.M. 2020. First-wave COVID-19 transmissibility and severity in China outside Hubei after control measures, and second-wave scenario planning: a modelling impact assessment. Lancet 395(10233): 1382–1393.

Michali. 2022. What is Black Box Testing? https://www.checkpoint.com/cyber-hub/cyber-security/what-is-penetration-testing/what-is-black-box-testing/#:~:text=Black%20box%20testing%2C%20a%20form,other%20aspects%20of%20an%20application.

Patel, S., Nargunde, R., Shah, C. & Dholay, S. 2021. Counterfeit Currency Detection using Deep Learning. 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT).

Rutkowska, M. 2023. Use Case Diagrams: An Introduction - Altkom Software. https://www.altkomsoftware.com/blog/use-case-diagrams-an-introduction/.

Sabat, Salih & Muhamad, Sabat. (2022). Counterfeit Currency Recognition Using Deep Learning: A Review.

Software Requirements Specification (SRS): Definition, Example, How to Write, & More. 2024. . https://www.inflectra.com/Ideas/Topic/Requirements-Definition.aspx.

Staff, C. 2023. What Is UI Design? Definition, Tips, Best Practices. https://www.coursera.org/articles/ui-design.

Systems, O. (n.d.). Design Specifications (DS) | Ofni Systems. https://www.ofnisystems.com/services/validation/design-specification/.

Tedja, R.T. 2019. Software Development Model: Incremental Model. https://sis.binus.ac.id/2019/07/02/software-development-model-incremental-model/.

Terra, J. 2023. What is Client-Server Architecture? Everything You Should Know. https://www.simplilearn.com/what-is-client-server-architecture-article#:~:text=The%20client%2Dserver%20architecture%20refers,model%20or%20client%20server%20network.

UML Sequence Diagram Tutorial. (n.d.). . https://www.lucidchart.com/pages/uml-sequence-diagram#:~:text=Sequence%20diagrams%20are%20a%20popular,function%20before%20the%20lifeline%20ends.

Upadhyaya, A., Shokeen, V. & Srivastava, G. 2018. Analysis of Counterfeit Currency Detection Techniques for Classification Model. 2018 4th International Conference on Computing Communication and Automation (ICCCA).

Upadhyaya, A., Shokeen, V. & Srivastava, G. 2018. Analysis of Counterfeit Currency Detection Techniques for Classification Model. 2018 4th International Conference on Computing Communication and Automation (ICCCA).

Literature review. 2023. . https://www.ed.ac.uk/institute-academic-development/study-hub/learning-resources/literature-review.

What Are Real User Needs and How to Define Them? | Uptech. 2024. . https://www.uptech.team/blog/how-to-define-users-needs.

What is Object Detection? | IBM. (n.d.). . https://www.ibm.com/topics/object-detection.

Writing Design Specifications: Quick Guide — RewiSoft. 2024. . https://rewisoft.com/blog/how-to-write-the-design-specifications-quick-guide/.

Dr. Zainal Rasyid Mahayuddin
Fakulti Teknologi & Sains Maklumat
Universiti Kebangsaan Malaysia