Identifikasi Anggota TWICE dalam Citra dan Video Menggunakan Metode YOLO

Haikal Assyauqi (13522052) [1,2]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]13522052@std.stei.itb.ac.id, [2]haikalassyauqi70@gmail.com

***Abstract*—Perkembangan teknologi membawa banyak perubahan pada bidang hiburan. Makalah ini bertujuan untuk membangun sebuah program yang mengidentifikasi anggota *K-pop girl group* TWICE dalam format video maupun gambar menggunakan *You Only Look Once* (YOLO). Program ini menggunakan *dataset* dengan anotasi poligon untuk menangani tantangan saling menutupi ketika koreografi dinamis. Diharapkan, model ini dapat membantu penggemar baru untuk dapat mengenali anggota grup serta menjadi referensi dalam *human recognition* pada musik video yang kompleks.**

***Keywords*—YOLO, Deteksi objek, TWICE, Video, CNN.**

I. PENDAHULUAN

Industri hiburan Korea Selatan (K-Pop) telah berkembang secara pesat, dengan TWICE sebagai salah satu ikon utamanya. Sebagai salah satu cara mempromosikan grup K-Pop, banyak stasiun televisi Korea Selatan yang menayangkan program musik, selain menampilkan penampilan grup, stasiun TV juga menyediakan format "Fancam" di mana satu kamera hanya tertuju pada satu anggota secara langsung selama pertunjukkan, namun karena pergerakan kamera masih dikendalikan secara manual dan banyaknya anggota sebuah grup, tidak jarang anggota lain yang disorot kamera.

Permasalahan ini menjadi persoalan menarik di bidang pemrosesan citra digital untuk mendeteksi anggota grup K-Pop menggunakan algoritma *You Only Look Once* (YOLO), algoritma ini dipilih karena keunggulan dalam melakukan deteksi objek dengan cepat tanpa mengorbankan akurasi. Makalah ini bertujuan untuk mengimplementasi dan mengevaluasi kinerja YOLO dalam mendeteksi Sembilan anggota dari grup TWICE baik dalam media berbentuk gambar dan video.

II. LANDASAN TEORI

A. TWICE

TWICE merupakan grup K-Pop wanita yang beranggotakan 9 orang di bawah naungan JYP Entertainment, grup ini memulai debut pada tahun 2015 melalui program *Sixteen,* grup ini beranggotakan Im Nayeon, Yoo Jeogyeon, Hirai Momo, Minatozaki Sana, Park Jihyo, Myoui Mina, Kim Dahyun, Son Chaeyoung, dan Chou Tzuyu, dalam *dataset*, 9 anggota TWICE ini direpresentasikan sebagai kelas yang akan diidentifikasi oleh YOLO.

Sebagai subjek dari makalah ini, anggota TWICE memilki Tingkat kemiripan yang cukup tinggi dikarenakan pada saat perilisan lagu, baju yang dikenakan anggota TWICE cenderung mirip sehingga dapat menyulitkan model ketika membedakan, selain itu sebagai salah satu grup yang terkenal dengan kemampuan menarinya, TWICE sering melakukan perpindahan saat melakukan koreografi, hal ini memberikan oklusi parsial (bagian tubuh tertutup anggota lain) hingga membuat model semakin sulit dalam melakukan identifikasi.

Gambar II.1 Gambar Grup K-Pop TWICE
Sumber: https://en.wikipedia.org/wiki/File:Twice_-_Feel_Special.png

B. Citra

Citra merupakan sebuah *array* angka dalam bidang 2D yang secara khusus diatur dalam baris dan kolom. Citra didefinisikan sebagai intensitas cahaya pada bidang 2D dengan fungsi f(x,y) di mana x dan y adalah koordinat bidang 2D. Berdasarkan jenis warna, citra dapat dikelompokkan menjadi citra biner, yaitu citra yang hanya memiliki piksel 0 dan 1, citra *grayscale,* citra yang memiliki piksel dari 0 hingga 255, di mana 0 adalah hitam dan 255 adalah putih, sementara citra RGB, citra yang

memiliki 3 warna, merah, hijau, dan biru, yang setiap warna memiliki intensitas 0 hingga 255.

Citra dapat dibagi berdasarkan dimensi waktu, yaitu citra diam (*still image*), citra ini hanya memiliki dimensi spasial dalam bentuk dua dimensi, dan sifatnya statis, citra tidak berubah terhadap perubahan waktu, dan citra bergerak (*moving images*), selain memiliki dimensi spasial, citra ini memiliki dimensi waktu, dengan fungsi intensitas f(x,y,t), citra ini memiliki sifat dinamis, dimana citra berubah setiap perubahan waktu.

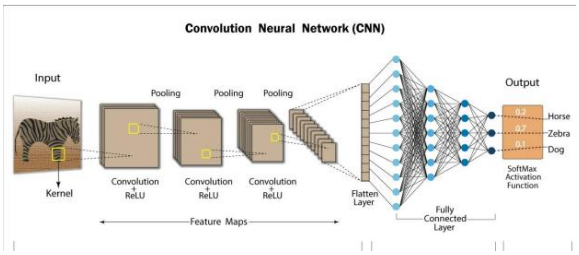
C. Video

Video merupakan salah satu contoh dari citra bergerak, video memiliki banyak *frame* dan dapat memiliki audio. Sebagai sebuah citra bergerak, video sebenarnya terbentuk dari berbagai kumpulan citra diam yang disebut dengan *frame,* yang ditampilkan dengan kecepatan tertentu yang disebut *Frame Rate* atau *Frame Per Second* atau biasa disingkat FPS, untuk menciptakan ilusi bahwa video merupakan citra yang bergerak, video biasanya memiliki kecepatan sekitar 24-60 FPS.

Selain *Frame Rate,* video juga memiliki resolusi spasial yang menunjukkan dimensi lebar dan tinggi piksel dalam video, semakin tinggi resolusi, semakin banyak detail yang bisa ditangkap.

D. Convolutional Neural Network (CNN)

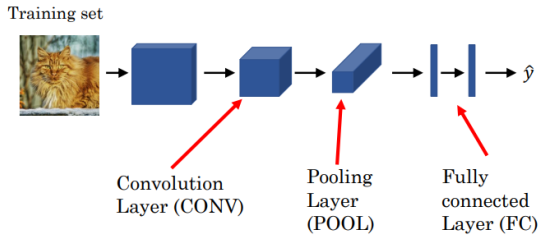
Convolutional Neural Network (CNN atau ConvNet) adalah algoritma pembelajaran mendalam yang populer, umumnya digunakan untuk memproses data yang memiliki topologi seperti grid. Contoh data berbentuk grid adalah citra. CNN merupakan arsitektur jaringan untuk pembelajaran mendalam yang belajar langsung dari data, dengan menghilangkan kebutuhan untuk melakukan ekstraksi fitur secara manual. CNN dapat disebut juga jaringan syaraf tiruan yang melibatkan konvolusi (CNN = ANN + convolution).



Gambar II.2 Arsitektur CNN
Sumber: Slide kuliah

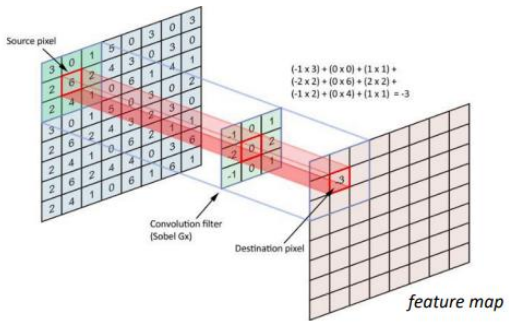
CNN memiliki 3 lapisan utama yaitu

1. *Convolutional Layer* (+ReLU)
2. *Pooling Layer*
3. *Fully-Connected Layer*



Gambar II.3 Lapisan Utama CNN
Sumber: Slide Kuliah

1. *Convolution Layer* (+ReLU)
   Convolutional Layer melakukan operasi konvolusi pada citra masukan dengan sejumlah penapis (filter). Tiap penapis menghasilkan luaran yang disebut *feature map.*

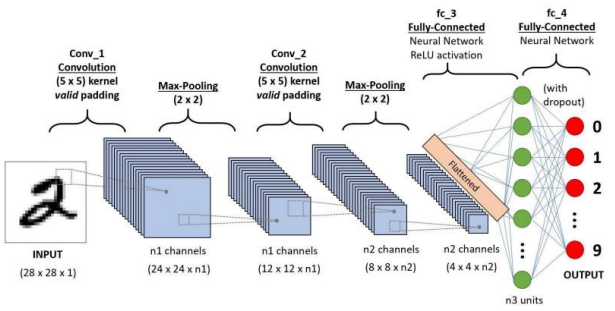


Gambar II.4 Cara Kerja *Convolution Layer*
Sumber: Slide Kuliah

   ReLU adalah layer tambahan yang memungkinkan pelatihan yang lebih cepat dan efektif dengan memetakan nilai negatif ke nol dan mempertahankan nilai positif. Pada dasarnya ReLU adalah operasi per-pixel dengan cara mengganti nilai negatif pixel di dalam *feature map* menjadi nol.

2. *Pooling Layer*
   Mirip dengan Convolutional Layer, Pooling Layer bertanggung jawab untuk mengurangi ukuran spasial dari matriks fitur hasil konvolusi. Hal ini bertujuan untuk mengurangi daya komputasi yang diperlukan untuk memproses data melalui pengurangan dimensi.

3. *Fully Connected Layer*
   Lapisan terakhir di dalam CNN adalah fully connected layer (FC) yang menghasilkan vektor dimensi K, dalam hal ini K adalah jumlah kelas yang dapat diprediksi oleh jaringan. Vektor ini berisi probabilitas untuk setiap kelas dari setiap gambar yang diklasifikasikan ..



Gambar II.5 Cara Kerja CNN
Sumber: Slide Kuliah

Untuk menghitung performa dari CNN, menggunakan *confusion matrix* untuk mengukur akurasi dengan rumus sebagai berikut

		Predicted class		
		yes	*no*	Total
Actual class	*yes*	*TP*	*FN*	*P*
	no	*FP*	*TN*	*N*
	Total	P'	N'	$P+N$

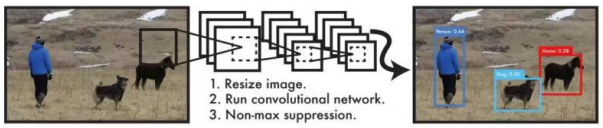
true positive (TP), true-negative (TN), false positive (FP), false negative (FN)

$$Akurasi = \frac{TP+TN}{TP+TN+FP+FN}$$

Gambar II.6 *Confusion Matrix* dan Akurasi
Sumber: Slide Kuliah

E. You Only Look Once (YOLO)

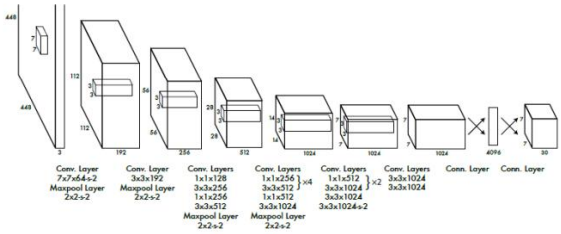
YOLO adalah algoritma untuk mendeteksi objek di dalam citra secara waktu nyata (*real time*). Objek yang dideteksi di dalam citra dibingkai di dalam bounding box beserta kelasnya. Disebut YOLO atau *You Only Look Once* karena YOLO melakukan hanya sekali pemindaian pada citra untuk mendeteksi dan mengidentifikasi objek yang terdapat di dalamnya. Oleh karena itu YOLO diklasifikasikan sebagai model dengan pendekatan *single-stage* atau *one-shot*. YOLO memiliki kecepatan prediksi yang tinggi sehingga cocok digunakan untuk keperluan pendeteksian objek secara *real-time*. YOLO dasar mampu memroses 45 *frame* per detik.



Gambar II.7 Hubungan YOLO dengan CNN
Sumber: Slide Kuliah

Cara kerja YOLO yaitu:

1. Citra masukan dilewatkan melalui CNN untuk mengekstraksi fitur-fitur citra tersebut.
2. Fitur-fitur tersebut kemudian dilewatkan melalui serangkaian *fully connected layer*, yang memprediksi probabilitas kelas dan koordinat kotak pembatas (*bounding box*).
3. Citra dibagi menjadi grid-grid, dan setiap piksel di dalam grid bertanggung jawab untuk memprediksi sekumpulan kotak pembatas (*bounding box*) dan probabilitas kelas.
4. Luaran jaringan adalah sekumpulan kotak pembatas dan probabilitas kelas untuk setiap sel.
5. Kotak pembatas kemudian difilter menggunakan algoritma pascapemrosesan yang disebut *non-max supression* untuk menghilangkan kotak yang tumpang tindih dan memilih kotak dengan probabilitas tertinggi.
6. Luaran akhir adalah sekumpulan kotak pembatas yang diprediksi dan label kelas untuk setiap objek dalam citra.



Gambar II.8 Arsitektur Utama YOLO
Sumber: Slide Kuliah

Metrik yang umum digunakan untuk arsitektur YOLO ada 4 yaitu *Precision, Recall, mAP50,* dan *mAP50-95. Precision* merupakan kemampuan model dalam memprediksi apakah model sesuai dengan data uji, *Recall* merupakan kemampuan model menyadari bahwa ada kelas yang ada di dalam citra, *mAP50* adalah jumlah *bounding box* yang beririsan 50% dengan *bounding box* asli, *mAP50-95* merupakan pengembangan *Map50* dengan irisan 50-95 persen.

III. IMPLEMENTASI SOLUSI

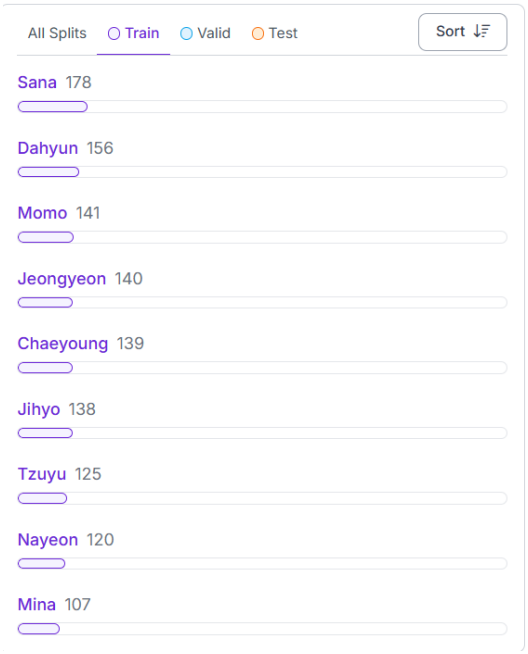
A. Pembangunan Dataset

Karena terbatasnya dataset yang berfokus pada setiap member TWICE, dataset dibuat menggunakan aplikasi Roboflow untuk membuat anotasi poligon, untuk datanya sendiri merupakan foto anggota TWICE ketika merilis lagu *Feel Special,* foto diambil dari musik video, penampilan di pertunjukkan musik, foto album, serta *vlog* dari kegiatan mereka selama proses pembuatan musik video (*Behind the* Scene), gambar dibagi menjadi 9 kelas sesuai jumlah anggota yaitu Nayeon, Jeongyeon, Sana, Momo, Jihyo, Mina, Dahyun, Chaeyoung, dan Tzuyu.

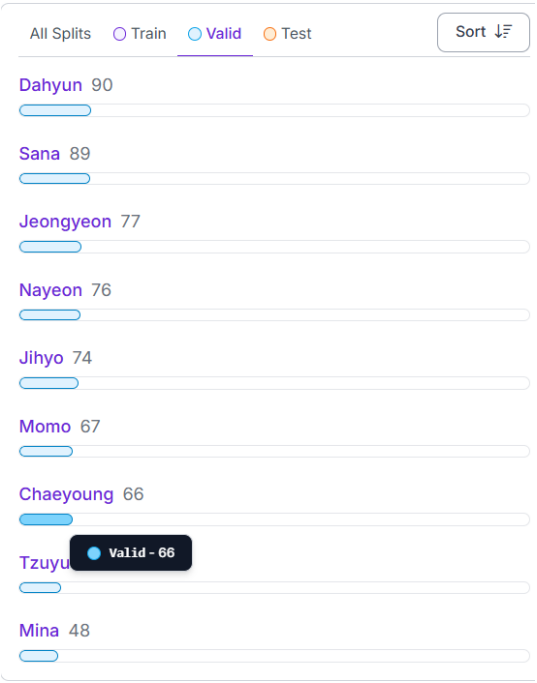
Gambar III.1 Anotasi Dataset
Sumber: https://universe.roboflow.com/haikal-

workspace/twice/analytics

Untuk penyebaran data sendiri setiap anggota memiliki gambar 100 hingga 180 gambar, pengecualian di sini adalah Mina, karena pada saat perilisan *Feel Special,* Mina mengalami masalah kesehatan sehingga data Mina cukup sedikit.



Gambar III.2 Penyebaran Kelas pada Data *Train*
Sumber: https://universe.roboflow.com/haikal-workspace/twice/analytics



Gambar III.3 Penyebaran Kelas pada Data *Valid*
Sumber: https://universe.roboflow.com/haikal-workspace/twice/analytics

B. Training

Untuk pelatihan menggunakan YOLO 11 Medium, model ini digunakan dikarenakan memiliki akurasi yang bagus serta cepat dan tidak menggunakan terlalu banyak sumber daya sehingga bisa digunakan laptop dengan GPU tingkat rendah, kelemahan dari model ini adalah, meskipun *input* dataset berupa poligon, namun model ini akan melakukan transformasi menjadi *bounding box,* sehingga keunggulan poligon dalam oklusi tidak terlalu berfungsi, karena dataset cukup kecil, ketika melakukan *training* gambar pada *dataset* banyak dilakukan manipulasi untuk memperbanyak variasi data untuk pelatihan, berikut adalah kode *training.*

```
from ultralytics import YOLO
if __name__ == "__main__":
    model = YOLO("yolo11m.pt")
    results = model.train(
        data="Dataset/data.yaml",
        epochs=300,
        patience=50,
        imgsz=640,
        batch=4,
        workers=2,
        pretrained=True,
        device=0,
        optimizer="AdamW",
        lr0=0.001,
        lrf=0.01,
        momentum=0.937,
        weight_decay=0.0005,
        warmup_epochs=3.0,
        warmup_momentum=0.8,
        warmup_bias_lr=0.1,
        box=7.5,
        cls=0.5,
        dfl=1.5,
        degrees=10.0,
        translate=0.1,
        scale=0.3,
        shear=2.0,
        perspective=0.0003,
        flipud=0.0,
        fliplr=0.5,
        hsv_h=0.015,
        hsv_s=0.7,
        hsv_v=0.4,
        mosaic=1.0,
        mixup=0.15,
        close_mosaic=15,
        copy_paste=0.1,
        dropout=0.1,
        label_smoothing=0.1,
        nms=True,
        iou=0.7,
        save=True,
        save_period=50,
    )
```

Beberapa parameter penting yang digunakan yaitu *epochs,* jumlah pelatihan, *patience,* ketika tidak ada peningkatan akan berhenti, *imgsz,* perubahan ukuran foto, *batch,* jumlah gambar yang diproses dalam satu iterasi, *optimizer,* algoritma optimasi yang digunakan, lalu augmentasi data seperti *degrees, shear, scale, translate, hsv, mosaic, mixup*. Hasil pelatihan ini akan mendapatkan sebuah model dengan format .pt yang akan digunakan untuk inferensi.

C. Inferensi

Usai mendapatkan model dengan format .pt, model bisa digunakan untuk inferensi dan prediksi, berikut adalah kode untuk melakukan prediksi dan inferensi, serta menampilkan hasil prediksi.

```
from ultralytics import YOLO
import cv2
import os

BASE_DIR = os.path.dirname(os.path.abspath(__file__))
model = YOLO(os.path.join(BASE_DIR, "runs/detect/train29/weights/best.pt"))
input_path = os.path.join(BASE_DIR, "inputImage/group4.jpg")
output_folder = os.path.join(BASE_DIR, "outputImage")
os.makedirs(output_folder, exist_ok=True)

COLORS = [
    (255, 0, 0),
    (0, 255, 0),
    (0, 0, 255),
    (255, 255, 0),
    (255, 0, 255),
    (0, 255, 255),
    (255, 128, 0),
    (128, 0, 255),
    (0, 128, 255)
]

def draw_label(img, text, pos, color):
    font = cv2.FONT_HERSHEY_SIMPLEX
    scale, thick, pad = 0.8, 2, 5
    (tw, th), _ = cv2.getTextSize(text, font, scale, thick)
    x, y = pos
    if y - th - pad * 2 < 0:
        y = th + pad * 2
    cv2.rectangle(img, (x, y - th - pad * 2), (x + tw + pad * 2, y), color, -1)
    brightness = color[0] * 0.114 + color[1] * 0.587 + color[2] * 0.299
    tc = (0, 0, 0) if brightness > 128 else (255, 255, 255)
    cv2.putText(img, text, (x + pad, y - pad), font, scale, tc, thick, cv2.LINE_AA)

def process_frame(img, results, show_conf=True):
    for r in results:
        best = {}
        boxes = r.boxes.xyxy.cpu().numpy()
        scores = r.boxes.conf.cpu().numpy()
        labels = r.boxes.cls.cpu().numpy()
        for box, score, label in zip(boxes, scores, labels):
            cid = int(label)
            if cid not in best or score > best[cid][1]:
                best[cid] = (box, score, model.names[cid])
        for cid, (box, score, name) in sorted(best.items(), key=lambda x: x[1][1]):
            color = COLORS[cid] if cid < len(COLORS) else (128, 128, 128)
            x1, y1, x2, y2 = map(int, box)
            cv2.rectangle(img, (x1, y1), (x2, y2), color, 3)
            label_text = f"{name} {score:.2f}" if show_conf else name
            draw_label(img, label_text, (x1, y1), color)
    return img

ext = os.path.splitext(input_path)[1].lower()
name = os.path.splitext(os.path.basename(input_path))[0]
IMAGE_EXT = ['.jpg', '.jpeg', '.png', '.bmp', '.tiff', '.webp']
VIDEO_EXT = ['.mp4', '.avi', '.mov', '.mkv', '.wmv', '.flv', '.webm']

if ext in IMAGE_EXT:
    print(f"Mode: GAMBAR - {input_path}")
    results = model.predict(source=input_path, imgsz=640, conf=0.1)
    img = process_frame(results[0].orig_img.copy(), results)
    save_path = os.path.join(output_folder, f"{name}_detected{ext}")
    cv2.imwrite(save_path, img)
    print(f"Tersimpan: {save_path}")

elif ext in VIDEO_EXT:
    print(f"Mode: VIDEO - {input_path}")
    cap = cv2.VideoCapture(input_path)
    fps = int(cap.get(cv2.CAP_PROP_FPS))
    w = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    h = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    total = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
```

```
        save_path = os.path.join(output_folder, f"{name}_detected.mp4")
        fourcc = cv2.VideoWriter_fourcc(*'mp4v')
        out = cv2.VideoWriter(save_path, fourcc, fps, (w, h))
        print(f"FPS: {fps}, Resolusi: {w}x{h}, Total: {total} frames")
        fc = 0
        while cap.isOpened():
            ret, frame = cap.read()
            if not ret:
                break
            fc += 1
            results = model.predict(source=frame, imgsz=640, conf=0.1, verbose=False)
            out.write(process_frame(frame, results, False))
            if fc % 30 == 0:
                print(f"Progress: {fc / total * 100:.1f}%")
        cap.release()
        out.release()
        print(f"Tersimpan: {save_path}")

    else:
        print(f"Format tidak didukung: {ext}")
```

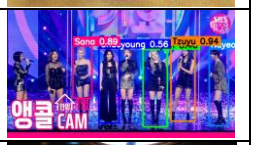
Kode ini dirancang untuk memproses masukan (*input*), baik berupa gambar maupun video, guna mengidentifikasi anggota TWICE berdasarkan model yang telah dilatih. Selain itu terdapat beberapa kode yang berfungsi untuk melakukan manipulasi agar tidak terlalu mengganggu hasil, sebagai contoh terdapat kode yang menghapus nilai optimistik pada keluaran video agar tidak ada perubahan nilai yang sering sehingga mengganggu tampilan video.

IV. Hasil Dan Analisis

Hasil dari *training* menggunakan YOLO11 Medium sebagai berikut

Class	Images	Instances	Box(P	R	mAP50	mAP50-95):
all	265	639	0.835	0.728	0.804	0.704
Chaeyoung	62	66	0.663	0.657	0.7	0.61
Dahyun	80	90	0.91	0.677	0.805	0.71
Jeongyeon	70	77	0.893	0.662	0.781	0.722
Jihyo	70	74	0.865	0.778	0.846	0.739
Mina	45	48	0.776	0.75	0.806	0.7
Momo	62	67	0.917	0.819	0.872	0.751
Nayeon	74	76	0.9	0.763	0.864	0.753
Sana	74	89	0.81	0.753	0.786	0.694
Tzuyu	50	52	0.781	0.692	0.776	0.656

Untuk hasil dan analisis, akan digunakan anggota TWICE dalam format gambar dan video, untuk format gambar akan diberikan 5 data tes dan untuk video akan diberikan 2 video berupa musik video dan *dance practice*. Berikut hasil percobaan dengan program:

No.	*Input*	*Output*
1		
2		
3		
4		
5		

Dari hasil analisis, dari 5 gambar, 3 gambar dapat dideteksi dengan baik, untuk 2 gambar yang tidak terlalu baik yakni gambar 2 dan gambar 4, untuk gambar 2, 6 anggota bisa dideteksi dengan baik dan 2 anggota tidak terdeteksi, untuk gambar 4, 3 anggota terdeteksi dengan baik, 2 anggota terdeteksi salah, dan 3 anggota tidak terdeteksi. Untuk gambar 2, hasil prediksi kurang akurat karena gambar tersebut merupakan gambar yang diambil melalui *screenshot* ketika anggota TWICE sedang *perform,* hal ini menyebabkan gambar blur yang membuat model kesulitan untuk memprediksi, untuk gambar 4, model cukup buruk dalam memprediksi karena dataset yang digunakan banyak mengandung anggota TWICE ketika menggunakan baju cerah, sehingga kesulitan dalam mengenali anggota, selain itu posisi anggota ketika foto ini diambil tidak keseluruhannya dalam posisi menghadap kamera, sehingga model yang dilatih dengan dataset anggota menghadap kamera tidak terlalu baik.

Karena tidak bisa memasukkan video ke dalam dokumen, berikut tautan untuk video sebagai berikut: Video 1 masukan memiliki tautan https://youtu.be/yChy1KRIWAY?si=OmQb72QfT1lpJs4x dan keluaran deteksi https://youtu.be/JMKBFsX1sl0, serta video 2 masukan memiliki tautan https://youtu.be/3ymwOvzhwHs?si=GQEwZzKplTAtEHTO dan keluaran deteksi https://youtu.be/BFDP89uVMoc

Dari video ini dapat disimpulkan bahwa video yang

selalu menampilkan 9 anggota secara konsisten memiliki akurasi yang jauh lebih baik dibandingkan dengan video yang memiliki perubahan anggota secara tiba-tiba, selain itu *background* pada video 2 jauh lebih ramai dibandingkan video 1 sehingga sering terjadi salah deteksi *beckground* menjadi anggota TWICE.

V. Kesimpulan Dan Saran

Dari sini didapatkan bahwa diperlukan dataset yang lebih banyak agar pengenalan anggota tidak hanya bergantung pada baju atau rambut dari anggota grup, selain itu untuk pengembangan berikutnya dapat menggunakan model yang dapat mengonsumsi masukan berupa poligon, bukan *bounding box.* dan juga agar model tidak kebingungan dengan *background*, perlu juga untuk melakukan anotasi untuk *background,* agar model bisa mempelajari hal lain selain anggota grup.

Tautan Video Youtube

https://youtu.be/FaoD6Svj0X0

Ucapan Terima Kasih

Pertama-tama, penulis mengucapkan terima kasih kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, penulis bisa menyelesaikan tugas makalah ini dengan baik. Penulis juga mengucapkan terima kasih kepada Bapak Rinaldi Munir selaku dosen mata kuliah Interpretasi dan Pengolahan Citra, yang selama ini telah membimbing dalam pembelajaran Interpretasi dan Pengolahan Citra. Selain itu, penulis juga berterima kasih kepada subjek makalah ini yaitu TWICE yang menemani penulis begadang ketika mengerjakan tugas Pengolahan Citra Digital maupun pengerjaan proposal tugas akhir.

References

[1] https://universe.roboflow.com/haikal-workspace/twice. Diakses pada 20 Desember 2025

[2] https://kprofiles.com/twice-members-profile/. Diakses pada 22 Desember 2025.

[3] https://docs.ultralytics.com/. Diakses pada 22 Desember 2025.

[4] https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2025-2026/01-Pengantar-Pemrosesan-Citra-Digital-Bag1-2025.pdf. Diakses pada 23 Desember 2025

[5] https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2025-2026/21-CNN-2025.pdf. Diakses pada 24 Desember 2025

[6] https://informatika.stei.itb.ac.id/~rinaldi.munir/Citra/2025-2026/24-YOLO-2025.pdf. Diakses pada 24 Desember 2025

Pernyataan

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 24 Desember 2025

Haikal Assyauqi
13522052