

Implementasi CNN (*Convolutional Neural Network*) Untuk Mendeteksi Penggunaan Masker Menggunakan Open CV

Muhammad Derio^{1*}, Hendri Ardiansyah¹

¹Fakultas Ilmu Komputer, Teknik Informatika, Universitas Pamulang, Jl. Raya Puspiptek No. 46, Kel. Buaran, Kec. Serpong, Kota Tangerang Selatan. Banten 15310, Indonesia

Email: ^{1*}mrmderi1997@gmail.com, ²dosen00832@unpam.ac.id

(* : coressponding author)

Abstrak– Penggunaan teknologi khususnya seperti visi komputer sangat dibutuhkan untuk beberapa bidang tertentu. Penerapannya yang luas bisa diterapkan dalam berbagai bidang seperti kedokteran, keamanan, dan otomotif untuk mobil kendali otomatis. Penggunaan visi komputer yang luas diharapkan bisa memberikan keuntungan khususnya bagi manusia. Pendeteksi penggunaan masker ini dikembangkan dengan bantuan pustaka OpenCV dan metode CNN. Proses pertama yang diberikan adalah mempelajari model data yang perlu dipelajari melalui algoritma pembelajaran dalam. Setelah data dipelajari langkah selanjutnya data tersebut diklasifikasikan berdasarkan pembagian kelompok atau biasa disebut sigmoid dalam pembelajaran dalam. Hasil akhir yang diberikan ialah memberikan suatu prediksi.

Kata Kunci: Visi Komputer, OpenCV, CNN, Pembelajaran Dalam

Abstract– The use of technology, especially computer vision, is needed for certain fields. Its wide application can be applied in various fields such as medicine, security, and automotive for automatic control cars. The use of broad computer vision is expected to provide benefits, especially for humans. Face mask detector was developed with the help of the OpenCV library and the CNN method. The first process given is to learn the data model that needs to be leaned through deep learning algorithms. After the data has been trained, the next step is to classify the data based on the division of groups or commonly called sigmoid in deep learning. The final result given is to provide a prediction.

Keywords: Computer Vision, OpenCV, CNN, Deep Learning

1. PENDAHULUAN

Perkembangan teknologi merupakan hal yang meningkatkan produktivitas bagi suatu bangsa. Berbagai bidang dan sub bidang ilmu juga terus berkembang dengan berbagai temuan baru. Hal tersebut diharapkan bisa memberikan suatu kontribusi yang baik bagi masyarakat. Dilihat dari perspektif kebudayaan, bahwa perkembangan teknologi juga sangat berpengaruh terhadap kebudayaan dan berbagai aspek dalam kehidupan. Perluasan teknologi diharapkan bisa meningkatkan taraf hidup bagi masyarakat.

Visi komputer sendiri memiliki pengertian yaitu merupakan sebuah disiplin ilmu yang berhubungan dengan sistem buatan bagaimana computer menerjemahkan atau memberikan informasi dari suatu citra atau gambar dan bisa juga melalui video. Visi komputer sendiri merupakan bidang yang dibutuhkan untuk membantu bidang lain dalam memberikan solusi atas permasalahan tertentu. Prinsip kerja dari visi komputer adalah mengambil informasi dari beberapa piksel pada gambar. Dalam suatu piksel tersebut berisi informasi yang dapat diterjemahkan melalui konsep perwarnaan dalam grafis biasa disebut RGB (*Red, Green, Blue*).

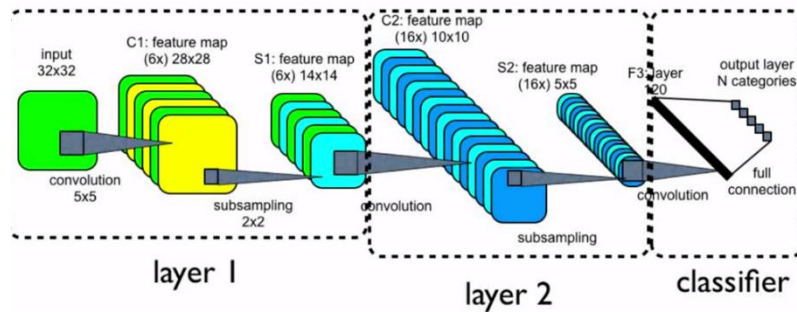
Proses kerja visi komputer sendiri sangat berbeda dengan pengolahan citra pada dasarnya pada pengolahan citra sendiri lebih menekankan pada aspek pemrosesan suatu gambar menjadi gambar lain. Dan proses kerja pengolahan citra biasanya dimanipulasi dalam bentuk bit pada piksel untuk beberapa operasi tertentu misalkan seperti untuk membedakan objek pada gambar (recognition), dan membedakan pola (pattern recognition).

Proses klasifikasi gambar dengan metode CNN diharapkan bisa menyelesaikan masalah dalam mendeteksi keberadaan masker dengan bantuan serta konsep dari aplikasi visi komputer. Karena fungsi utama CNN sendiri adalah mempelajari lapisan setiap gambar dari gambar dengan binary klasifikasi (binary classification). Sehingga hasil akhir memberikan model berupa kriteria dari bit yang dipelajari tadi.

2. METODOLOGI PENELITIAN

2.1 Metode CNN

Metode CNN merupakan Kelas jaringan saraf tiruan (neural network), yang digunakan untuk mengimplementasikan pengklasifikasian gambar. Menurut (Ian Goodfellow, Yoshua Bengio, 2016) *convolutional neural network* banyak diaplikasikan mengambil pendekatan yang berbeda terhadap regulasi, convolutional neural network memanfaatkan pola hierarki dalam data dan mengumpulkan pola kompleksitas yang meningkat menggunakan pola yang lebih kecil dan lebih sederhana yang timbul dalam filter.



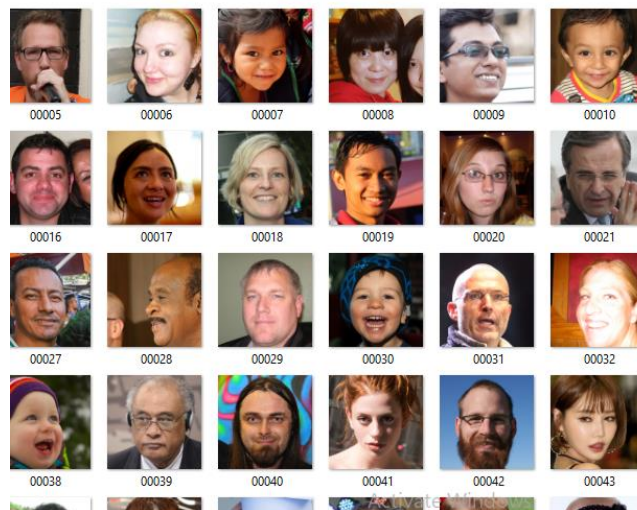
Gambar 1. Metode *Convolutional Neural Network*

Input file adalah sebuah sample file sebelum ditraining, fungsi sub-sampling sendiri mengurangi ukuran sample dari misalkan berukuran 1024x1024 per sample gambar kemudian di downsampling atau di resize menjadi 224x224 piksel. Kemudian pengkelasan gambar dibagi menjadi beberapa kategori. Dengan bantuan Pustaka tensorflow pengkategorian bisa dibagi menjadi beberapa subjek. Misalkan dengan memilih metode pengeklasan dengan sigmoid membaginya menjadi 2 kelas. Yaitu salah satu kelas bernilai 1 dan 0.

3. ANALISA DAN PEMBAHASAN

Dalam penelitian ini dibutuhkan beberapa proses modelling dengan bantuan Pustaka keras dan tensorflow. Hasil prediksi pada penelitian ini tidak selalu menciptakan hasil yang konstan artinya bersifat berubah – ubah artinya hasil prediksi bersifat variable dari satu kurun waktu pembelajaran model.

Komponen yang digunakan adalah dataset seperti dibawah ini:



Gambar 2. Dataset Yang Digunakan

Pembelajaran dataset bisa dilakukan dengan proses pre-trained model dengan mempelajari 1755 sample dataset, prosesnya seperti dibawah ini:

```

Settings Pengklasifikasian Gambar dan Binary dari gambar

In [53]: new_model.compile(loss="binary_crossentropy", optimizer="adam", metrics= ["accuracy"])

In [*]: new_model.fit(X, Y, epochs=1, validation_split=0.1)

WARNING:tensorflow:From C:\Program Files\Python37\lib\site-packages\tensorflow_core\python\ops\math_grad.py:1424: where (from tensorflow.python.ops.array_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
Train on 1755 samples, validate on 195 samples
1024/1755 [=====] - ETA: 1:51 - loss: 0.3401 - acc: 0.9727
    
```

Gambar 3. Proses Modelling Dataset

Hasil proses training model bisa dilihat dibawah ini
 Model: "model"

Tabel 1. Hasil Pengujian Dari Dataset Yang Di Train

Layer (type)	Output Shape	Param
input_1 (InputLayer)	[(None, 224, 224, 3)]	0
conv1_pad (ZeroPadding2D)	(None, 225, 225, 3)	0
conv1 (Conv2D)	(None, 112, 112, 32)	864
conv1_bn (BatchNormaliza	(None, 112, 112, 32)	128
conv1_relu (ReLU)	(None, 112, 112, 32)	0
conv_dw_1 (DepthwiseConv2D)	(None, 112, 112, 32)	288
conv_dw_1_bn (BatchNormaliza	(None, 112, 112, 32)	128
conv_dw_1_relu (ReLU)	(None, 112, 112, 32)	0
conv_pw_1 (Conv2D)	(None, 112, 112, 64)	2048
conv_pw_1_bn (BatchNormaliza	(None, 112, 112, 64)	256
conv_pw_1_relu (ReLU)	(None, 112, 112, 64)	0
conv_dw_2 (DepthwiseConv2D)	(None, 56, 56, 64)	576
conv_dw_2_bn (BatchNormaliza	(None, 56, 56, 64)	256
conv_dw_2_relu (ReLU)	(None, 56, 56, 64)	0
conv_pw_2 (Conv2D)	(None, 56, 56, 128)	8192
conv_pw_2_bn (BatchNormaliza	(None, 56, 56, 128)	512
conv_pw_2_relu (ReLU)	(None, 56, 56, 128)	512
conv_pw_2_relu (ReLU)	(None, 56, 56, 128)	512
conv_dw_3 (DepthwiseConv2D)	(None, 56, 56, 128)	1152
conv_dw_3_bn (BatchNormaliza	(None, 56, 56, 128)	512
conv_dw_3_relu (ReLU)	(None, 56, 56, 128)	0
conv_pad_4 (ZeroPadding2D)	(None, 57, 57, 128)	0
conv_dw_2 (DepthwiseConv2D)	(None, 57, 57, 128)	0
conv_pw_4_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_pw_4_relu (ReLU)	(None, 28, 28, 256)	0
conv_dw_5_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_dw_5_relu (ReLU)	(None, 28, 28, 256)	0
conv_pw_5 (Conv2D)	(None, 28, 28, 256)	65536
conv_pw_5_bn (BatchNormaliza	(None, 28, 28, 256)	1024
conv_pad_6 (ZeroPadding2D)	(None, 29, 29, 256)	0
conv_dw_6 (DepthwiseConv2D)	(None, 14, 14, 256)	2304
conv_dw_6_bn (BatchNormaliza	(None, 14, 14, 256)	1024
conv_dw_6_relu (ReLU)	(None, 14, 14, 256)	0
conv_dw_7 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_7_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_7_relu (ReLU)	(None, 14, 14, 512)	0

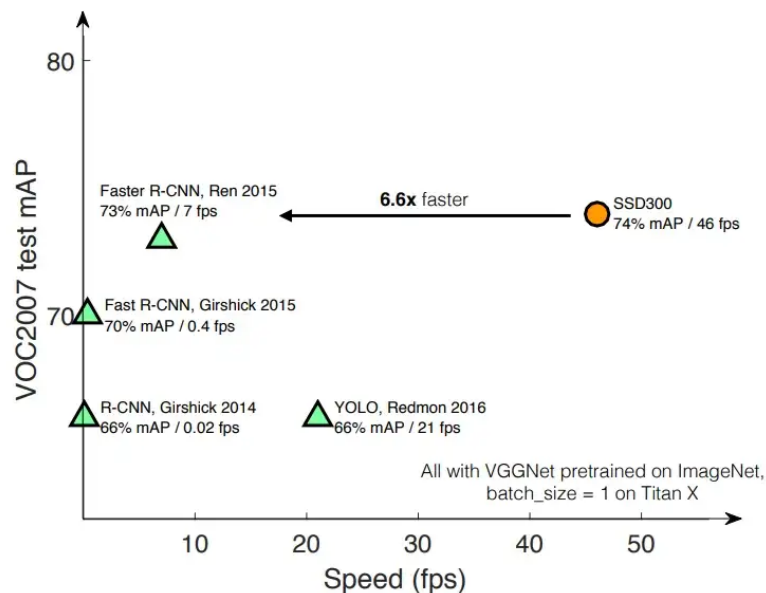
conv_dw_8 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_8_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_8_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_9 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_9_bn (BatchNormaliza	(None, 14, 14, 512)	2048
conv_dw_9_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_10 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_10_bn (BatchNormaliz	(None, 14, 14, 512)	2048
conv_dw_10_relu (ReLU)	(None, 14, 14, 512)	0
conv_dw_11 (DepthwiseConv2D)	(None, 14, 14, 512)	4608
conv_dw_11_bn (BatchNormaliz	(None, 14, 14, 512)	2048
conv_dw_11_relu (ReLU)	(None, 14, 14, 512)	0
conv_pad_12 (ZeroPadding2D)	(None, 15, 15, 512)	0
conv_dw_12 (DepthwiseConv2D)	(None, 7, 7, 512)	4608
conv_dw_12_bn (BatchNormaliz	(None, 7, 7, 512)	2048
conv_dw_13 (DepthwiseConv2D)	(None, 7, 7, 1024)	9216
conv_dw_13_bn (BatchNormaliz	(None, 7, 7, 1024)	4096
conv_dw_13_relu (ReLU)	(None, 7, 7, 1024)	0
conv_pw_13 (Conv2D)	(None, 7, 7, 1024)	1048576
global_average_pooling2d	(Gl (None, 1024)	0
reshape_1 (Reshape)	(None, 1, 1, 1024)	0
dropout (Dropout)	(None, 1, 1, 1024)	0
flatten (Flatten)	(None, 1024)	0
dense (Dense)	(None, 1)	1025

Total params : 3,229,889

Trainable params : 3,208,001

Non-trainable params: 21,888

Dari segi performa CNN lebih baik dari segi performa dan kecepatan dibandingkan metode YOLO (*You Look Only Once*) dalam pembelajaran (*training*) model.



Gambar 4. Perbandingan Performa CNN dengan YOLO

Sample yang dipelajari kemudian diuji hasil pengujiannya untuk gambar sample dibagi menjadi 2 sample tidak bermasker dan bermasker, seperti dibawah ini:



Gambar 5. Sample Tidak Bermasker

Pada sample tidak bermasker diperoleh nilai prediksi bernilai positif + dan dicapai 36 % akurasi bahwa objek tersebut tidak mengenakan masker.



Gambar 6. Sample Bermasker

Pada sample bermasker diperoleh nilai prediksi bernilai negative dan dicapai 36 % akurasi bahwa objek tersebut mengenakan masker.

4. IMPLEMENTASI

Proses implementasi sendiri dilakukan dengan beberapa langkah. Proses cukup lama karena jumlah data yang dipelajari atau bisa disebut training sangat banyak. Data yang berjumlah banyak tersebut biasa disebut dataset. Bahkan sebuah dataset bisa memiliki jumlah kumpulan data gambar dengan kategori tertentu bisa mencapai 130.000 lebih gambar. Banyaknya data tersebut yang diolah tentu membutuhkan spesifikasi komputer yang sangat tinggi dengan kemampuan dari segi processing serta kinerja yang stabil.

Spesifikasi perangkat lunak pada proses penelitian:

Sistem Opearasi : Windows 11 Pro 64-bit

Web Browser : Mozilla Firefox

IDE : Jupyter Notebook

Library : Tensorflow OpenCV

Spesifikasi perangkat keras pada proses penelitian:

Processor : Intel® Core™ i7-6600U CPU @ 2.60GHz (4 CPUs)

Memory : 8 GB

Storage : 250GB SSD

DirectX Version : DirectX 12

Pengujian sistem dilakukan pertama adalah dengan proses modelling setelah itu data dipelajari dan dianalisis sesuai dengan kriteria pembelajaran mesin yang dilakukan. Proses prediksi memberikan output bahwa jika mengenakan masker maka output bernilai negatif berdasarkan kriteria pendeteksian masker menggunakan klasifikasi binary (binary classification).

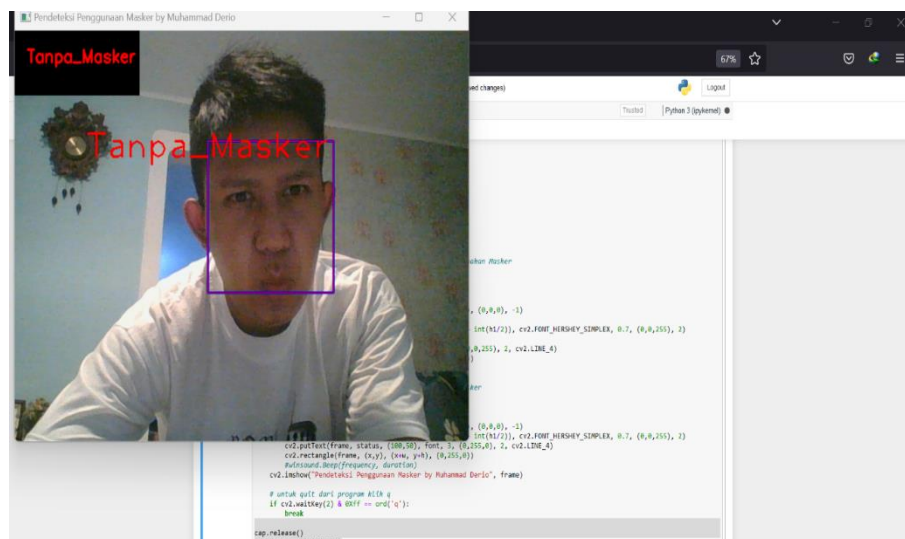
1. Hasil Implementasi Jika Mendeteksi Keberadaan Masker



Gambar 7. Hasil Implementasi Jika Mendeteksi Keberadaan Masker

Jika sistem mendeteksi keberadaan objek mengenakan masker maka hasil output akan memberikan output teks berwarna hijau. Output tersebut dipelajari melalui proses klasifikasi gambar yang dipelajari oleh metode CNN. Metode CNN membagi kedua kelompok tersebut menjadi beberapa nilai output. Nilai bernilai positif maka dinilai akan bernilai negatif.

2. Hasil Implementasi jika tidak mendeteksi keberadaan masker



Gambar 8. Hasil Implementasi Jika Tidak Mendeteksi Keberadaan Masker

Dibagian perbandingannya adalah jika hasil output mendeteksi objek tidak mengenakan masker maka akan memberikan teks berupa warna merah. Artinya adalah output bernilai positif ketika tidak mengenakan masker. Objek tersebut bernilai positif saat terjadi proses pengecekan masker. Dalam hasilnya output tidak selalu berjalan dengan akurat. Semuanya dipengaruhi oleh beberapa faktor seperti : tingkat pencahayaan ruangan, spesifikasi perangkat lunak serta spesifikasi perangkat keras. Banyak faktor lain seperti fungsi loss pada pembelajaran dalam yaitu Ketika mempelajari data ada probabilitas (kemungkinan) bahwa data bersifat error saat dipelajari. Hal tersebut juga memberikan kontribusi untuk tingkat keakuratan dalam proses pendeteksian suatu objek penggunaan masker.

5. KESIMPULAN

Berdasarkan hasil pengujian dan penelitian yang dilakukan pada penelitian ini dapat disimpulkan seperti berikut:

1. Penelitian Implementasi CNN (Convolutional Neural Network) untuk mendeteksi penggunaan masker menggunakan OpenCV dilakukan menggunakan Bahasa pemrograman python, dan penambahan API (Application Programming Interface) dari pustaka OpenCV. Dan dibantu dengan penambahan pustaka tensorflow untuk membantu proses modelling dataset.
2. Berdasarkan hasil penelitian bahwa output setelah di pelajari (training) yang menggunakan masker memiliki prediksi bernilai negatif sedangkan disisi lain output bernilai positif untuk yang bukan menggunakan masker. Hasil output dipengaruhi oleh beberapa faktor seperti pencahayaan menjadi faktor penting dan spesifikasi perangkat lunak serta perangkat keras saat melakukan modelling dalam proses pembelajaran dataset.

REFERENCES

- Ding, Yuchen; Li, Zichen; & Yastermsky, David. (2021). Real-time Face Mask Detection In Video Data. University of Pennsylvania. Arxiv: 2005.0395050v2.
- Huang, Zhibing & Liang, Jinbi. (2020). Masked Face Recognition Dataset And Application. Arxiv : 2003:09093v2
- Kadir, Abdul. (2019). Langkah mudah mempelajari OpenCV dan Python. Jakarta: Elex Media Komputindo
- Mohan, Puranjay; Jyoti, Aditya; & Chirania, Abhay. (2021). A Tiny A CNN Architecture For Medical Face Mask Detection For Resource – Constrained Endpoints. SRM Institute Of Science and Technology, Kattankulakhtur, Tamil Nadu, India. Arxiv:2011.14858v3.
- O' Mahony, Niall; Campbell, Sean; Carvalho, Anderson; Harapanhalli, Suman; Velasco, Gustavo; Krpalkova, Lenka; Riordan, Daniel & Walsh, Joseph. (2019). Deep Learning Versus Traditional Computer. Imap Technology Gateway. Institute Of Tralee, Ireland.
- Paullada, Amandalyne; Bender, Emily; Denton, Emily; & Hanna, Alex. (2021). A survey of dataset development and use in Machine learning. Science Direct.
- Triantafillou, Eleni; Zhu, Tyler; Dumoulin Vincent; Lamblin, Pascal; Evci, Utku; Xu, Kelvin; Goroshin, Ross. (2019). Meta-dataset : A dataset of datasets for leaning to learn from few examples. Arxiv:1903.03096
- Larson, Stefan; Mahendran, Anish; Peper, Joseph; Clarcke, Cristopher, Lee, Andrew; Hill, Parker; K, Jonathan. (2019). An Evaluation Dataset For Intent Classification And Out-Of-Scope Prediction. Arxiv: 1909.02027
- Northcutt, Curtis; Jiang, Lu; & Chuang, Isaac. (2021). Confident learning: Estimating Uncertainty In Dataset Labels. (2021). Journal Of Artificial Intelligence Research 70, 1373-1411
- Brdjanin, Adnan; Dardagan, Nadja; Dzigal, Dzemil; & Akagic, Amila (2020). Single Object Trackers in OpenCV : A Benchmark.