

Deep Learning untuk Mendeteksi Pola Kebotakan Rambut pada Pria dengan Metode CNN (*Convolutional Neural Network*)

Aji Pangestu dan Aries Muslim

Sistem Informasi Bisnis, Manajemen Sistem Informasi, Universitas Gunadarma
Jl. Margonda Raya 100, Depok Jawa Barat, Indonesia -16424
E-mail : apang471@gmail.com, aries.muslim09@gmail.com

Abstrak

Kerontokan rambut pada pria sering disebut juga sebagai *Male Pattern Baldness* menyebabkan penipisan pada rambut di area kepala pria hal ini terjadi karena erat kaitannya dengan hormon yang berperan penting yaitu testosteron dan *dihydrotestosterone* (DHT). Pada penderita *male pattern baldness*, kadar DHT ini lebih tinggi dari normal dan biasanya disertai reseptor pada folikel rambut yang memang lebih peka terhadap DHT, Namun *male pattern baldness* dapat diukur dengan pola tertentu yang dinamakan *Hamilton-Norwood Scale*. Pada penelitian ini akan menerapkan metode *Deep Learning* untuk mendeteksi pola kebotakan rambut yang dialami pria berdasarkan pola ukur *Hamilton-Norwood Scale* menggunakan *Tensorflow Object Detection API* dengan menerapkan *Convolutional Neural Network* (CNN) pada klasifikasi citra yang digunakan. Penelitian ini diharapkan dapat melakukan deteksi dan mengukur akurasi setiap klasifikasi yang telah ditentukan berdasarkan pola ukur yang hasilnya akan ditampilkan pada OpenCV (*Open Source Computer Vision Library*). Dataset yang dipergunakan pada penelitian ini dibuat dengan sejumlah 248 gambar yang dibagi kedalam tiga bagian, yaitu dataset *train*, dataset *test*, dan dataset *image testing* yang perbandingan data yang digunakan pada setiap dataset yaitu sebesar 65% : 20% : 15%. Proses pelatihan dilakukan dengan menggunakan Tensorflow GPU, Anaconda dan Jupyter Notebook. Penelitian ini menggunakan Python 3.6.3 sebagai bahasa pemrograman dan library Python. Dari hasil uji coba pada penelitian ini diperoleh tingkat akurasi sebesar 62,5% dari hasil akhir sebanyak 15 gambar tidak terdeteksi. Penelitian ini berhasil mendeteksi dan mengklasifikasi pola kebotakan rambut pada pria sesuai dengan pola *Hamilton-Norwood Scale*.

Kata kunci: Deep Learning, Deteksi Objek, Python, Tensorflow, CNN

Pendahuluan

Kerontokan rambut pada pria sering disebut juga sebagai *Male Pattern Baldness* ini erat kaitannya dengan hormonal, hormon yang berperan penting adalah *testosterone* dan *dihydrotestosterone* (DHT). *Testosterone* dirubah menjadi DHT oleh enzyme 5 alpha reductase. DHT ini memerikan ciri kelamin sekunder pada laki-laki berupa tumbuhnya kumis, jambang, bulu ketiak namun menyebabkan penipisan pada rambut di area kepala. Inilah sebabnya saat memasuki pubertas rambut halus didahi mulai menghilang dan hal tersebut merupakan hal yang wajar. Pada penderita *Male Pattern Baldness*, kadar DHT ini yang lebih tinggi dari normal dan juga disertai reseptor pada folikel rambut yang memang lebih peka terhadap DHT. Kadar DHT yang tinggi ini menyebabkan folikel rambut mengalami

miniaturisasi sehingga rambut tumbuh dengan ukuran lebih tipis disertai ukuran akar rambut yang lebih kecil [1].

Male Pattern Baldness pada umumnya dapat diukur dengan pola tertentu yang dinamakan *Hamilton Norwood Scale* dan untuk mengukur pola kebotakan ini biasanya pasien membutuhkan bantuan tenaga dokter atau membandingkan secara manual dengan pola Hamilton [2].

Dimasa sekarang kecerdasan buatan memiliki peran inti dalam membantu menyelesaikan banyak dari pekerjaan manusia. Dengan kecerdasan buatan komputer dapat diajarkan cara untuk mengenal model *neural network* dengan menyediakan data *input* dan *output* dalam proses pembelajarannya yang terarah yang dikenal sebagai *Supervised Learning*. Proses pembelajaran oleh mesin ini lebih

dikenal dengan nama *Machine learning* dan *Deep Learning*.

Secara sederhana *machine learning* membangun sebuah algoritma yang memungkinkan program komputer untuk belajar dan melakukan tugasnya sendiri tanpa diintruksikan lagi oleh pengguna. Pada *Deep learning* pembelajaran dilakukan dengan lebih banyak layer dari *machine learning*.

Proses pembelajaran yang mengajarkan bagaimana Komputer dapat melihat dikenal dengan istilah *Computer Vision*. Dengan *Computer Vision* mesin mampu untuk mengekstrak informasi dari data berupa gambar untuk menyelesaikan perintah tertentu. Teknik *Computer Vision* yang paling populer adalah pendeteksian objek.

Deteksi objek memungkinkan komputer untuk dapat mengenal objek dalam suatu gambar atau video. Salah satu *library* yang dikembangkan oleh google untuk mempelajari pembelajaran mesin dan deteksi objek adalah *tensorflow Object Detection API*. Pembelajaran yang telah berhasil akan memperlihatkan hasil objek berupa area yang terdeteksi dan akurasi.

Beberapa penelitian yang membahas mengenai *Deep learning* menggunakan *tensorflow* untuk *object detection* diantaranya adalah penelitian untuk mendeteksi tanda normor kendaraan bermotor menggunakan *tensorflow*. Penelitian tersebut, menggunakan algoritma CNN untuk mendeteksi tanda kendaraan bermotor pada sebuah gambar dengan jumlah 502 dataset, 80% training data dan 20% testing data. Menggunakan batch 4 dan dengan 100.000 steps Penelitian tersebut menghasilkan akurasi sekitar 99% [3].

Selanjutnya, penelitian yang menggunakan *object detection* untuk mendeteksi dan mengklasifikasi objek meja dan kursi bermotif ukiran jepara. Penelitian tersebut juga menggunakan algoritma CNN dan *tensorflow* untuk mendeteksi dan mengklasifikasi dua objek yaitu meja dan kursi motif ukiran jepara dengan jumlah 500 dataset, 80% training data dan 20% testing data. Menggunakan *batch* 2 dan dengan 250.000 *steps* penelitian tersebut berhasil menghasilkan akurasi 70%- 99% [4].

Penelitian yang menggunakan *Deep Learning* dengan model algoritma *Fast R-CNN* untuk mendeteksi tiga klasifikasi objek yaitu celana, baju, dan kacamata, penelitian tersebut menggunakan 4373 dataset. 3677 data train dan 696 data testing. Menghasilkan rata rata akurasi sebesar 78% untuk baju, untuk celana rata rata sebesar 65% dan untuk kacamata rata rata sebesar 57% [5].

Penelitian yang mengimplementasikan salah satu metode *Machine Learning* yang dapat digunakan untuk klasifikasi citra objek yaitu CNN. Penelitian tersebut menggunakan dua tahap. Tahap pertama adalah klasifikasi citra menggunakan *feedforward*. Tahap kedua pembelajaran menggunakan metode *backpropagation*. Sebelum dilakukan klasifikasi terlebih dahulu objek yang akan diklasifikasi difokuskan dengan metode *crop-*

ping dan *wrapping*. Selanjutnya dilakukan training menggunakan metode *feedforward* dan *backpropagation*. Hasil uji coba dari klasifikasi citra objek pada basis data Caltech 101 menghasilkan akurasi rata-rata sampai 100% [6].

Penelitian yang menggunakan *deep learning* untuk mendeteksi lampu lalu lintas dengan membandingkan dua model deteksi objek dan dengan mengevaluasi fleksibilitas *Tensorflow Object Detection Framework* untuk menyelesaikan masalah secara *real-time*. Dalam perbandingannya menggunakan *Single Shot Multibox Detector* (SSD) MobileNet V2 dan Faster-RCNN. Hasil pembelajaran memperoleh hasil bahwa *Faster-RCNN* menghasilkan 97%, dan SSD menghasilkan 58 % dari 441 dataset yang dilatih [7].

Berdasarkan latar belakang diatas, penelitian ini akan menggunakan Deep learning dengan metode CNN untuk pelatihan deteksi objek menggunakan *library Tensorflow*. Hasil pelatihan akan ditampilkan menggunakan OpenCV (*Open Source Computer Vision Library*) yang akan mendeteksi 4 objek dan menampilkan akurasi, 4 objek berdasarkan 4 pola *Hamilton Norwood scale*, yaitu Norwood1, Norwood2, Norwood3, dan pola akhir Norwood4567.

Pengolahan Citra

Pengolahan Citra Digital merupakan bidang ilmu yang mempelajari tentang bagaimana suatu citra itu dibentuk, diolah, dan dianalisis sehingga menghasilkan informasi yang dapat dipahami oleh manusia. Sedangkan Citra adalah gambar pada bidang dua dimensi dan disusun oleh banyak pixel yang merupakan bagian terkecil dari citra. Pada umumnya, citra dibentuk dari kotak-kotak persegi empat yang teratur sehingga jarak horisontal dan vertikal antara piksel sama pada seluruh bagian citra [8].

Machine Learning

Istilah *Machine Learning* didefinisikan pertama kali oleh Arthur Samuel pada tahun 1959. Menurut Arthur Samuel, *Machine learning* adalah cabang ilmu dari *Artificial Intelligence* yang memungkinkan komputer memiliki kemampuan untuk belajar tanpa perlu di program lagi. Secara sederhana *machine learning* membangun sebuah algoritma yang memungkinkan program komputer untuk belajar dan melakukan tugasnya sendiri tanpa adanya intruksi dari penggunanya [9].

Computer Vision

Istilah Computer Vision pertama kali didefinisikan oleh Shapiro dan Stockman pada tahun 2001. Menurut Shapiro dan Stockman, *Computer vision* merupakan salah satu cabang ilmu pengetahuan yang bertujuan untuk membuat suatu keputusan

yang berguna untuk mengenali objek fisik nyata dan keadaan berdasarkan sebuah gambar atau citra.

Computer Vision menjadikan komputer berfungsi seperti mata manusia, sehingga mendekati kemampuan manusia dalam menangkap informasi visual. Kemampuan itu seperti mengenali adanya objek pada sebuah gambar dan mengetahui dimana batasnya, menempatkan label pada objek, menafsirkan gerakan [10].

Object Detection

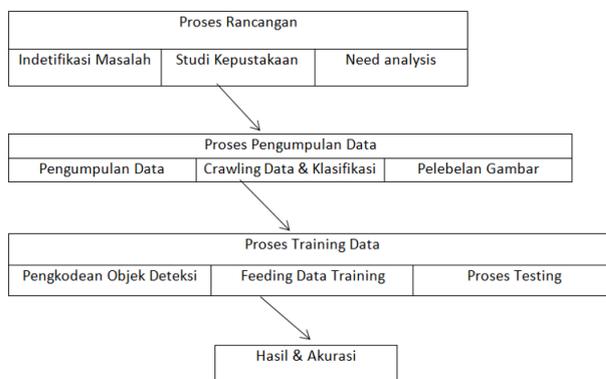
Object Detection merupakan teknik visi komputer untuk menemukan contoh objek dalam gambar atau video. Algoritma deteksi objek biasanya memanfaatkan pembelajaran mesin atau pembelajaran mendalam untuk menghasilkan hasil yang bermakna. Ketika manusia melihat gambar atau video, manusia dapat mengenali dan menemukan objek dalam beberapa saat berbeda dengan komputer yang memerlukan komputasi yang kompleks.

Tujuan deteksi objek adalah untuk mereplikasi kecerdasan yang dimiliki manusia dalam melihat benda menggunakan komputer. Cara kerja deteksi objek adalah menempatkan keberadaan objek dalam gambar dan menggambar kotak pembatas di sekitar objek tersebut. Ini biasanya melibatkan dua proses, yaitu mengklasifikasikan jenis objek, dan kemudian menggambar kotak disekitar objek tersebut.

Klasifikasi gambar dan skenario deteksi objek terlihat serupa. Secara umum, klasifikasi adalah mengklasifikasikan gambar ke dalam kategori tertentu. Sedangkan objek deteksi adalah mengidentifikasi lokasi objek dalam gambar, dan misalnya menghitung jumlah akurasi suatu objek [11].

Metode Penelitian

Penelitian ini memiliki serangkaian proses yang dilakukan untuk menentukan akurasi dan hasil, dapat dilihat pada Gambar 1.



Gambar 1: Metode Penelitian

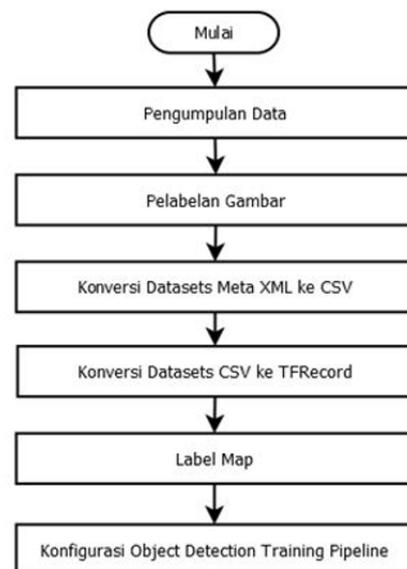
Pada penelitian proses pertama yang dilakukan adalah proses rancangan. Dimulai dengan indenti-

fikasi masalah, kemudian proses pembelajaran atau studi kepustakaan tentang deeplearning dan deteksi objek. Selanjutnya *need analysis* menentukan kebutuhan *hardware* dan *software* yang akan digunakan pada penelitian. Pada proses pengumpulan data, data dikumpulkan dan diklasifikasikan sesuai dengan klasifikasi yang ditentukan pada penelitian, kemudian dilakukan pelabelan pada data gambar yang sudah disesuaikan dengan klasifikasi yang ditentukan.

Pada proses training data, dilakukan pengkodean serta perancangan sistem untuk mendeteksi klasifikasi objek yang sudah ditentukan sehingga dapat diukur pada proses testing. Setelah semua proses telah selesai akan dilakukan perhitungan untuk menentukan hasil akurasi yang akan diperoleh pada penelitian.

Pengumpulan & Pelabelan Data

Penggunaan dataset pada metode CNN yaitu berupa data gambar. Model CNN akan berjalan dengan baik ketika menggunakan data train gambar yang banyak. Sehingga sebuah model dapat belajar mengenali gambar tersebut. Dataset yang digunakan dalam penelitian ini berupa gambar yang dikumpulkan melalui search engine google. Data gambar yang digunakan pada penelitian ini adalah gambar pria berambut normal dan pria berpola kebotakan. Dataset gambar yang telah dikumpulkan akan dilakukan pelabelan ,untuk lebih jelasnya alur proses pengumpulan data dapat dilihat pada Gambar 2.



Gambar 2: Alur Pengumpulan Data

Pelabelan gambar dilakukan pada objek gambar yang akan dideteksi sesuai dengan klasifikasi yang sudah ditentukan sebelumnya, gambar yang telah dilabel disimpan dalam bentuk file XML. Proses selanjutnya itu konversi data gambar dari XML ke

CSV sehingga diperoleh ukuran detail label gambar yang terbatas koma. Setelah melalui proses konversi ke CSV data gambar tersebut dilakukan konversi lagi ke TFRecord sehingga data label gambar tersebut bisa dilakukan pelatihan pada *Tensorflow* untuk proses training data.

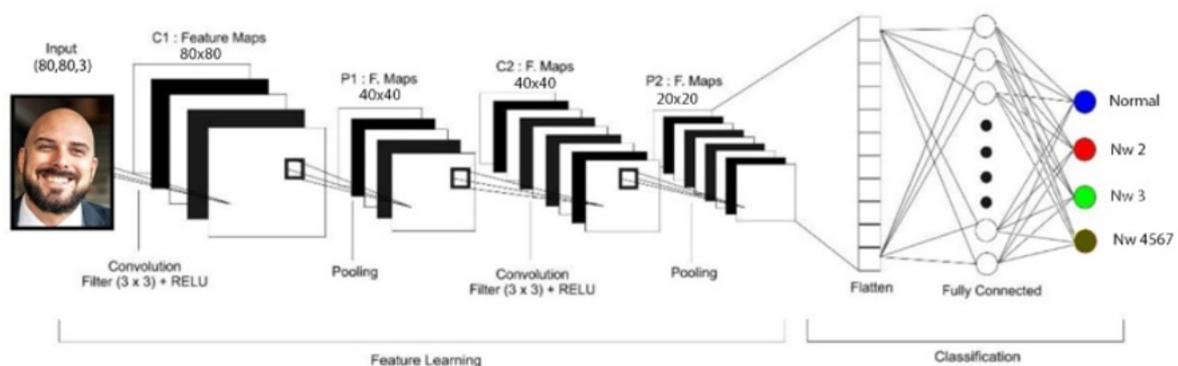
Proses Classification

Pada proses *classification* data gambar yang telah dilabel dan dikonversi ke TFRecord akan dilakukan proses Label Map untuk diberikan klasifikasi sesuai dengan jumlah klasifikasi yang ditentukan pada penelitian. Dalam penelitian ini peneliti menguji data gambar pola kebotakan pada pria dengan 4 klasifikasi, yang nantinya data gambar pria berpola kebotakan tersebut dapat dibedakan sesuai klasifikasi yang telah ditentukan pada proses Label Map

ini. Untuk efisiensi pada proses pengumpulan data ini dilakukan konfigurasi deteksi objek *pipeline* sehingga memungkinkan data untuk dilakukan proses training kembali.

Pemodelan Jaringan Convolutional Neural Network (CNN) & Training Data

Setelah melalui beberapa proses, yaitu dari proses pelabelan, konversi data maka selanjutnya data yang telah dikumpulkan akan ditraining dengan menggunakan algoritma CNN. Pembentukan model jaringan CNN sangat mempengaruhi hasil dari loss yang didapat dari model. pemodelan jaringan pada proses training dapat dilihat, pada Gambar 3.



Gambar 3: Model Jaringan

Dilihat dari Gambar 3, maka proses training data melewati beberapa tahapan layer yaitu layer *Convolution*, *Pooling*, dan *Fully Connected*. Pada layer konvolusi terjadi proses penyaringan. Seperti layaknya gambar, filter pada proses konvolusi ini memiliki ukuran karnel 3x3 dengan jumlah filter 80. Tahapan ini adalah proses kombinasi antara dua buah matriks yang berbeda untuk menghasilkan nilai matriks yang baru.

Setelah tahap konvolusi selesai, maka akan ditambahkan sebuah aktivasi baru yaitu RELU (*Rectified Linear Unit*). RELU ini berfungsi untuk mengubah nilai negatif menjadi nol (menghilangkan nilai negatif pada matriks hasil konvolusi). Nilai *padding* yang digunakan dalam tahapan konvolusi adalah 0, sehingga hasil ukuran dari konvolusi tetap sama yaitu 80x80.

Pada *pooling layer* terjadi proses pengurangan ukuran matriks dengan memakai operasi *pooling*. *Pooling layer* terdiri atas sebuah filter yang ukuran nilai matriksnya secara bergantian akan bergeser pada area *feature maps*. Hasil dari tahapan *pooling* akan menghasilkan nilai matriks yang baru, karena dalam penelitian ini akan menggunakan aktivasi *maxpooling*. Cara kerja dari *maxpooling* adalah dengan

mengambil nilai maksimum berdasarkan karnelnya.

Tahap konvolusi yang kedua yaitu meneruskan hasil dari tahapan *pooling* pertama, yaitu dengan *input* matriks gambar yang berukuran 40x40 dengan jumlah filter 80 yang ukuran karnelnya 3x3. Pada tahapan ini juga menggunakan aktivasi RELU.

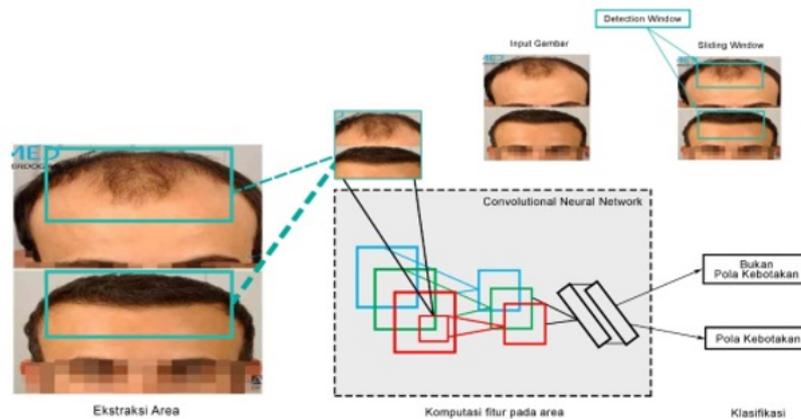
Tahapan selanjutnya yaitu *pooling* kedua, yang mana proses tahapannya sama dengan *pooling* yang pertama. Perbedaan antara keduanya hanya pada hasil akhir matriksnya. *Pooling* kedua menghasilkan *output* yang berukuran 40x40. Setelah itu masuk ke tahapan *flatten* atau *fully connected*. Tahapan ini hanya terdiri dari satu *hidden layer*.

Tahapan *flatten* mengubah *output pooling layer* menjadi vektor. Lalu selanjutnya tahapan *flatten* akan melakukan proses pendeteksian gambar. Tahapan yang terakhir adalah penggunaan aktivasi fungsi *softmax* pada *output layer*. Untuk menentukan klasifikasi hasil akhir proses training.

Rancangan Sistem

Rancangan sistem yang digunakan untuk melakukan deteksi objek terdiri dari masukan (*input*), pembelajaran (*training*) dan keluaran (*output*). Secara garis besar dapat dilihat dari Gambar 4.

(*put*), pembelajaran (*training*) dan keluaran (*output*). Secara garis besar dapat dilihat dari Gambar 4.

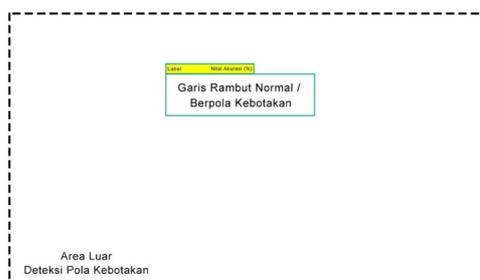


Gambar 4: Model Jaringan

Input yang digunakan untuk hasil uji coba berupa 208 gambar, kemudian gambar tersebut melalui tahapan pembelajaran yang ada pada model jaringan, setelah selesai proses training selesai kemudian akan dilakukan testing output dengan menggunakan 40 gambar baru untuk menentukan hasil dan akurasi. Rancangan Tampilan Hasil Deteksi Rancangan tampilan hasil deteksi akan menggunakan library OpenCV untuk membantu menampilkan output hasil deteksi. Dapat dilihat pada Gambar 5.

Tabel 1: Proses Training

Waktu(menit)	Total_Loss	Classification_Loss	Objectness_Loss
1	1.423	0.3948	0.1806
5	1.289	0.2799	0.01186
10	0.1988	0.09077	0.01206
15	0.1020	0.01743	0.01539



Gambar 5: Tampilan Hasil Deteksi

Pada Tabel 1 hasil training berjalan dengan baik dan akan dilanjutkan dalam waktu 8 jam untuk mendapatkan hasil yang optimal. Proses selanjutnya yaitu tahapan testing, setelah dilakukan training selama 8 jam hasil dan akurasi dari deteksi objek dapat dilihat pada Tabel 2.

Tabel 2: Hasil Testing

Testing Kelas Normal				
No.	Gambar	Type	Testing Score	Object Detect
1.		.jpg	Normal 96% Kecepatan 144ms	true
2.		.png	Tidak terdeteksi	false
3.		.jpg	Normal 98% Kecepatan 153ms	true
4.		.png	Normal 96% Kecepatan 155ms	true
5.		.jpg	Normal 98% Kecepatan 151ms	true
6.		.jpg	Normal 88% Kecepatan 164ms	true

Hasil dan Pembahasan

Setelah selesai melakukan proses training yang menggunakan batch size 1, epoch sebesar 1 dan learning rate 0,002. Dilakukan proses monitoring selama 15 menit proses training berlangsung melalui tensorboard (grafik visualisasi proses training pada tensorflow) dan didapatkan hasil yang dapat dilihat pada Tabel 1.

7.		.jpg	Normal 96% Waktu proses 144ms	true
8.		.png	Normal 88% Waktu proses 160ms	true
9.		.png	Tidak terdeteksi	false
10.		.png	Normal 78% Waktu proses 167ms	true

Testing Kelas Norwood 2

No.	Gambar	Type	Testing Score	Object Detect
1.		.jpg	Norwood2 97% Kecepatan 123ms	true
2.		.png	Norwood2 88% Kecepatan 169ms	true
3.		.jpg	Norwood2 98% Kecepatan 150ms	true
4.		.png	Tidak terdeteksi	false
5.		.jpg	Norwood2 98% Kecepatan 154ms	true
6.		.png	Tidak terdeteksi	false
7.		.jpg	Norwood2 96% Waktu proses 124ms	true
8.		.png	Tidak terdeteksi	false
9.		.jpg	Tidak terdeteksi	false
10.		.png	Norwood2 78% Waktu proses 167ms	true

Testing Kelas Norwood 3

No.	Gambar	Type	Testing Score	Object Detect
1.		.jpg	Norwood3 97% Kecepatan 123ms	true
2.		.png	Norwood3 88% Kecepatan 169	true
3.		.jpg	Norwood3 98% Kecepatan 150ms	true
4.		.png	Tidak terdeteksi	false
5.		.jpg	Norwood3 98% Kecepatan 154ms	true
6.		.png	Tidak terdeteksi	false
7.		.jpg	Norwood3 96% Waktu proses 124ms	true
8.		.png	Tidak terdeteksi	false
9.		.jpg	Tidak terdeteksi	false
10.		.png	Tidak terdeteksi	false

Testing Kelas Norwood4567

No.	Gambar	Type	Testing Score	Object Detect
1.		.jpg	Norwood4567 97% Kecepatan 129ms	true
2.		.png	Norwood4567 86% Kecepatan 150ms	true
3.		.jpg	Norwood4567 98% Kecepatan 150ms	true
4.		.png	Norwood4567 99% Kecepatan 170ms	true
5.		.jpg	Norwood4567 98% Kecepatan 154ms	true
6.		.png	Tidak terdeteksi	false
7.		.jpg	Norwood4567 96% Waktu proses 124ms	true
8.		.png	Tidak terdeteksi	false
9.		.jpg	Tidak terdeteksi	false
10.		.png	Tidak terdeteksi	false

Dengan menggunakan total 40 gambar maka diperoleh rincian hasil dari 4 kelas yang sudah diklasifikasi maka akan dinilai apakah terdeteksi atau tidaknya gambar tersebut, untuk rinciannya dapat dilihat pada Tabel 3.

Tabel 3: Rincian Testing

No.	Kelas Kebotakan	Jumlah Data Uji	Tidak terdeteksi	Terdeteksi
1	Normal	10	2	8
2	Norwood2	10	4	6
3	Norwood3	10	5	5
4	Norwood4567	10	4	6
Total		40	15	25

Dari rincian tabel testing maka dapat dilakukan perhitungan akurasi. Perhitungan akurasi dilakukan untuk menilai persentase keberhasilan model CNN dalam mengklasifikasi gambar. Akurasi juga dapat menjadi tolak ukur dalam perbandingan maupun pengembangan model CNN hasil training ini. proses perhitungan akurasi menggunakan persamaan

$$Akurasi = \frac{Jumlah\ Data\ Benar}{Jumlah\ Data\ Uji} \times 100\%$$

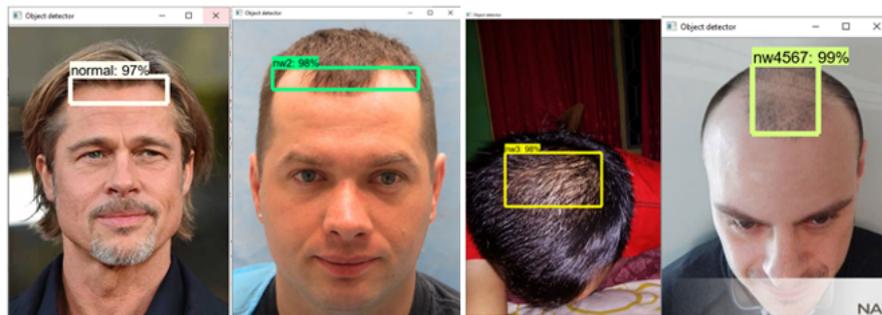
$$Sehingga, Akurasi = \frac{25}{40} \times 100\% = 62,5\%$$

Hasil akurasi sudah cukup mendapatkan hasil yang optimal dalam pendeteksiannya, untuk akurasi pada *openCV library* hasil deteksi menunjukkan 70% - 99% pada objek hasil deteksi.

Output Hasil Deteksi

Penelitian ini berhasil mendeteksi dan mengklasifikasi pola kebotakan rambut pada pria sesuai den-

gan *Hamilton Norwood Scale* dengan akurasi yang sangat baik pada aplikasinya yang ditampilkan oleh OpenCV dapat dilihat pada Gambar 6.



Gambar 6: Output Hasil Deteksi

Penutup

Berdasarkan penelitian yang telah dilakukan, proses pendeteksian pola kebotakan rambut pada pria menggunakan algoritma CNN berhasil dideteksi dengan cukup baik dan berhasil untuk mengklasifikasi setiap kelas pada proses uji coba. Dari 40 data testing 25 gambar berhasil dideteksi dengan tampilan akurasi 70% - 99%. Sehingga diperoleh rata-rata akurasi keberhasilan sejumlah 62,5%.

Penelitian ini dapat dikembangkan dengan menambahkan dataset training dan data uji coba. Pengembangan dapat dilanjutkan dengan metode yang berbeda. Selain itu, kemungkinan pada implementasinya penelitian ini dapat digunakan untuk membantu pria yang bermasalah dengan kerontokan rambut dapat mengetahui lebih awal tentang tahapan kerontokan rambutnya sehingga dapat dicegah perkembangannya. Serta dapat membantu dokter spesialis kulit dan kelamin untuk memperkenalkan *Hamilton Norwood Scale* pada pasien sehingga mempermudah pekerjaannya.

Daftar Pustaka

- [1] Mrinal Gupta and Venkataram Mysore, "Classifications of Patterned Hair Loss: A Review", *Journal of Cutaneous and Aesthetic Surgery*, 9(3)., DOI: 10.4103/0974-2077.178536 , 2016.
- [2] Koo SH, Chung HS, Yoon ES, Park SH, "A new classification of male pattern baldness and a clinical study of the anterior hairline", *Aesthetic Plast Surg*, 24:46-51, 2000.
- [3] Imam Taufiq, "Deep Learning Untuk Deteksi Tanda Nomor Kendaraan Bermotor Menggunakan Algoritma Convolutional Neural Network Dengan Python Dan Tensorflow", Skripsi Program Studi Sistem Informasi Sekolah Tinggi Manajemen Informatika dan Komputer AKAKOM, 2018.
- [4] S. Ravi dan S. Nayeem, "A Study on Face Recognition Technique based on Eigenface," *International Journal of Applied Information Systems*, vol. 5, no. 4, pp. 57-62, 2013
- [5] Helder Filipe de Sausa Russa, "Computer Vision: Object Recognition With Deep Learning Applied to Fashion Items Detection in Images", Tesis. Faculdade de Economia Universidade Do Porto. 47, 2017.
- [6] Eka Putra dan Wayan Suartika, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101", *Jurnal Teknik ITS*, 5, 2016.
- [7] T.V. Janahiraman and M.S.M. Subuhan, "Traffic light detection using tensorflow object detection framework", *International Conference on System Engineering and Technology (ICSET)*, 2019.
- [8] A.F. Firiya, M.I. Irawan dan Soetrisno, "Implementasi Extreme Learning Machine untuk Pengenalan Objek Citra Digital", *Jurnal Sains dan Seni ITS*. Vol 6, No.1, 2017.
- [9] S. R. Dewi, "Deep learning object detection pada video menggunakan tensorflow dan convolutional network," Skripsi, Universitas Islam Indonesia, Yogyakarta, 2018.
- [10] Muhammad Zufar dan Budi Setiyono, "Convolutional neural networks untuk pengenalan wajah secara real-time". *Jurnal Sains dan Seni ITS*, Vol. 5 No. 2, Institut Teknologi Sepuluh Nopember (ITS), 2016.
- [11] Prisky Ratna Aningtiyas, Agus Sumin dan Setia Wirawan, "Pembuatan Aplikasi Deteksi Objek Menggunakan TensorFlow Object Detection API dengan Memanfaatkan SSD MobileNet V2 Sebagai Model Pra - Terlatih", *Jurnal Ilmiah KOMPUTASI*, Vol. 19, No. 3, 2020.