

Abstract

Video surveillance is important in automatic surveillance. In the search for pedestrians when the face image can no longer be recognized, then what can be recognized is the attribute of the pedestrian. Therefore we need a system that can recognize the attributes of pedestrians that are expected to help in video surveillance. In this study a pedestrian attribute recognition system was built using the Convolution Neural Network (CNN) method so that it can search for pedestrians with certain attributes. In recognizing pedestrian attributes, the authors conducted a study using transfer learning from the Resnet50 architecture by using several scenarios in determining the threshold. The best results of this research indicate the average accuracy of 90.82% with data test.

Keywords: convolution neural network, pedestrian attributes, video surveillance, automatic surveillance

1. Pendahuluan

Latar Belakang

Dalam pengawasan video dunia nyata memanfaatkan manusia dalam melakukan pengawasan terhadap video seperti CCTV. Namun ketika manusia melakukan pengawasan secara terus menerus akan terjadi kelelahan yang mengakibatkan penurunan konsentrasi, terlebih ketika mengawasi banyak CCTV. Pengawasan video juga sering digunakan untuk pencarian pelaku [1] kriminal yang baru saja terjadi. Dalam banyak skenario pengawasan dunia nyata, untuk mengenali seseorang biasanya melalui pengenalan wajah [2], namun ketika citra wajah memiliki resolusi yang rendah sehingga wajah akan sulit bahkan tidak dapat dikenali. Maka tampilan seluruh tubuh adalah informasi visual yang dapat digunakan yaitu *Soft-Biometric* atau atribut pejalan kaki seperti ciri-ciri pakaian dan aksesoris yang digunakan sehingga pejalan kaki dapat dikenali melalui *Soft-Biometric* yang digunakannya.

Dalam melakukan pencarian dibutuhkan 2 tahap yaitu deteksi objek pejalan kaki dan pengenalan *Soft-Biometric* pejalan kaki. Dalam pengenalan *Soft-Biometric* pejalan kaki, terdapat beberapa tantangan yaitu, pengenalan *Soft-Biometric* merupakan masalah *multi-label* [3] yang lebih rumit daripada *single-label*. Kemudian dataset yang tersedia selalu memiliki keragaman penampilan, sudut pandang pengambilan citra, tingkat kecerahan citra yang beragam dan *imbalance label*.

Dalam pengenalan *Soft-Biometric* pejalan kaki, beberapa penelitian telah dilakukan. Mulai dari ekstraksi ciri secara *handcrafted features* seperti HOG [4], *texture feature* dan beberapa penggunaan *classifier* seperti Support Vector Machine (SVM) [5], K-Nearest Neighbour (K-NN), dan MRF [6]. Di sisi lain dalam beberapa tahun terakhir, salah satu dari metode *deep learning* yaitu CNN telah mencapai kinerja yang mengesankan karena keberhasilan pada ekstraksi fitur dan klasifikasi menggunakan jaringan konvolusi, terutama pada *computer vision* [9].

Maka diperlukan sebuah sistem yang dapat melakukan pengenalan *Soft-Biometric* pejalan kaki. Pada penelitian sebelumnya, kasus ini diselesaikan menggunakan *handcrafted feature extraction* dan klasifikasi SVM dan KNN, dengan kinerja yang masih kurang baik. Kinerja ini bisa ditingkatkan menggunakan CNN. CNN dapat melakukan ekstraksi fitur menggunakan jaringan konvolusi dan juga dapat digunakan sebagai klasifikasi. Oleh karena itu, pada penelitian ini diajukan metode CNN untuk meningkatkan kinerja pengenalan *Soft-Biometric* pejalan kaki.

Fokus dari penelitian ini adalah pada pengenalan *Soft-Biometric* pejalan kaki. Bagian selanjutnya, dijelaskan kajian pustaka yang berkaitan dengan penelitian ini, penjelasan metode yang digunakan, hasil dan analisis dari penelitian serta kesimpulan yang didapat dari penelitian yang dilakukan.

Topik dan Batasannya

Pengenalan *soft-biometric* pejalan kaki termasuk dalam topik *computer vision*. Pada penelitian sebelumnya, kasus ini diselesaikan menggunakan *handcrafted feature extraction* dan klasifikasi SVM dan KNN dengan kinerja yang masih kurang baik. Kinerja ini bisa ditingkatkan menggunakan CNN. Fokus utama pada penelitian ini adalah pada pengenalan *soft-biometric* pejalan kaki.

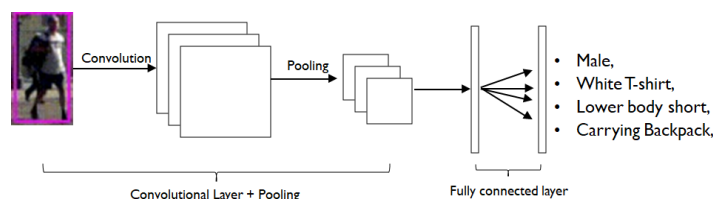
Tujuan

CNN dapat mengekstraksi fitur menggunakan jaringan konvolusi dan juga dapat digunakan sebagai klasifikasi. Oleh karena itu, pada penelitian ini menggunakan metode CNN untuk meningkatkan kinerja pengenalan *Soft-Biometric* pejalan kaki.

2. Studi Terkait

2.1 Convolutional Neural Network

Convolutional Neural Network (CNN) [7] adalah metode dalam *Deep Learning*. Metode tersebut memiliki kemampuan *feature engineering* yang dapat merencanakan fitur secara otomatis sehingga lebih mudah daripada membangun model ekstraksi fitur secara manual. Setelah melakukan ekstraksi fitur, metode ini juga dapat digunakan sebagai *classifier*. Sehingga metode tersebut banyak digunakan pada bidang *computer vision* seperti *image classification* [7], *image recognition* [7]. Secara garis besar struktur CNN terdiri dari lapisan masukan (*input layer*), sejumlah *hidden layer*, dan *output layer*. *Hidden layer* pada CNN umumnya terdiri dari beberapa jenis lapisan, yaitu *convolutional layers*, dan *ReLU layer*.



Gambar 1. Contoh arsitektur CNN

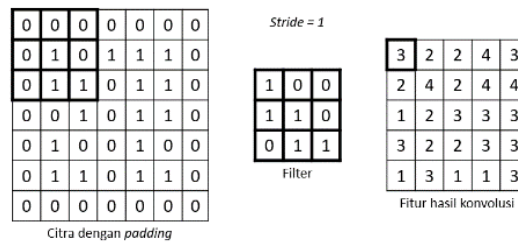
2.1.1 Convolutional Layer

Convolutional layer merupakan lapisan yang berfungsi melakukan ekstraksi ciri dari sebuah citra. Pada sebuah convolution layer pada umumnya terdiri dari padding, filter dan stride. Padding adalah parameter yang menentukan jumlah piksel yang akan ditambahkan pada setiap sisi dari masukan citra. Hal ini digunakan untuk memanipulasi dimensi feature map. Tujuan dari padding untuk mengatur dimensi output agar tetap sama seperti dimensi input atau setidaknya tidak berkurang secara drastis. Sehingga bisa menggunakan convolution layer yang lebih untuk mendapatkan banyak fitur yang diekstrak.

0	0	0	0	0	0	0
0	1	0	1	1	1	0
0	1	1	0	1	1	0
0	0	1	0	1	1	0
0	1	0	0	1	0	0
0	1	1	0	1	1	0
0	0	0	0	0	0	0

Gambar 2. Citra dengan padding

Konvolusi pada dasarnya berupa dot product antara filter yang berupa metrik dengan citra. Proses dot product akan bergeser sebesar stride hingga seluruh bidang telah selesai kemudian akan menghasilkan feature map hasil dari konvolusi seperti pada gambar 3.



Gambar 3. Proses konvolusi dengan filter 3x3 dan stride 1

2.1.2 Rectified Linear Units

Rectified Linear Units (ReLU) mengaplikasikan fungsi aktivasi $f(x) = \max(0, x)$. Hal ini biasa digunakan untuk meningkatkan sifat nonlinearitas fungsi keputusan dan jaringan secara keseluruhan tanpa mempengaruhi bidang-bidang reseptif pada convolution layer.

2.1.3 Pooling Layer

Pooling layer digunakan untuk mereduksi dimensi suatu citra atau matrik diteruskan kelapisan selanjutnya. Tujuan dari pooling layer untuk merepresentasikan data menjadi lebih kecil agar mudah dikelola, dan mengurangi proses komputasi. Terdapat dua jenis pooling layer, yaitu max pooling dan average pooling. Proses pooling yang umum digunakan adalah max pooling, memilih nilai maksimum dalam suatu area piksel tertentu (seluas filter). Sedangkan average pooling yaitu merata-ratakan suatu area piksel tertentu (seluas filter).

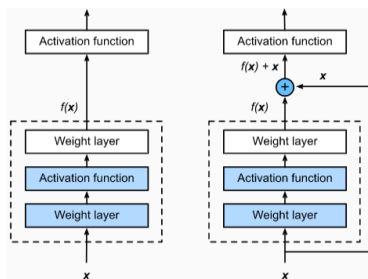


Gambar 4. Teknik max pooling (atas) dan average pooling (bawah)

2.2 Residual Network

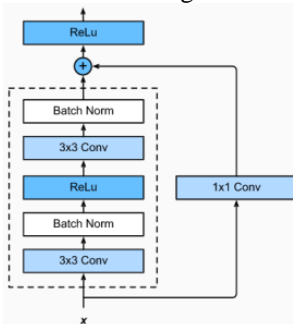
Residual Network (ResNet) merupakan arsitektur CNN yang berhasil meraih juara pertama dalam *ImageNet Large Scale Visual Recognition Competition 2015 (ILSVRC 2015)*, arsitektur tersebut dibuat oleh Kaiming He dkk[8]. untuk menangani *vanishing gradient problem* yang terjadi pada arsitektur CNN yang memiliki kedalaman tinggi. *Vanishing gradient problem*[8] adalah suatu keadaan dimana hasil gradien dari proses belajar jaringan, tidak dapat mencapai lapisan pertama karena mengalami penyusutan nilai yang disebabkan perkalian berkali-kali sehingga layer pertama tidak menerima gradien. Hal ini menyebabkan CNN tidak mendapat informasi untuk mengupdate bobot dari error yang dikalkulasikan.

Untuk menangani *vanishing gradient problem*, Kaiming He dkk[8]. mengusulkan penggunaan *skip connection* untuk membantu nilai gradien kembali tersebar. *Skip connection* adalah penambahan suatu nilai tertentu sebelum perhitungan fungsi aktifasi. *Skip connection* tersebut dapat melewati 1 atau lebih fungsi aktifasi, berbeda dengan *plain network* (CNN pada umumnya) yang tidak terdapat *skip connection* seperti pada gambar 5.



Gambar 5. Plain network (kiri) dan skip connection (kanan)

Setiap satu *skip connection* dikelompokkan kedalam 1 blok yang dinamakan *residual block* seperti pada ilustrasi gambar 6. Dalam *residual block* terdiri dari *convolution*, *batch normalization*, *ReLU* dan 1 jaringan yang bertugas sebagai *skip connection* yaitu *convolution*. Penggunaan *batch normalization* berfungsi untuk menormalisasikan hasil *convolution* sebelum masuk ke fungsi aktifasi ReLU.



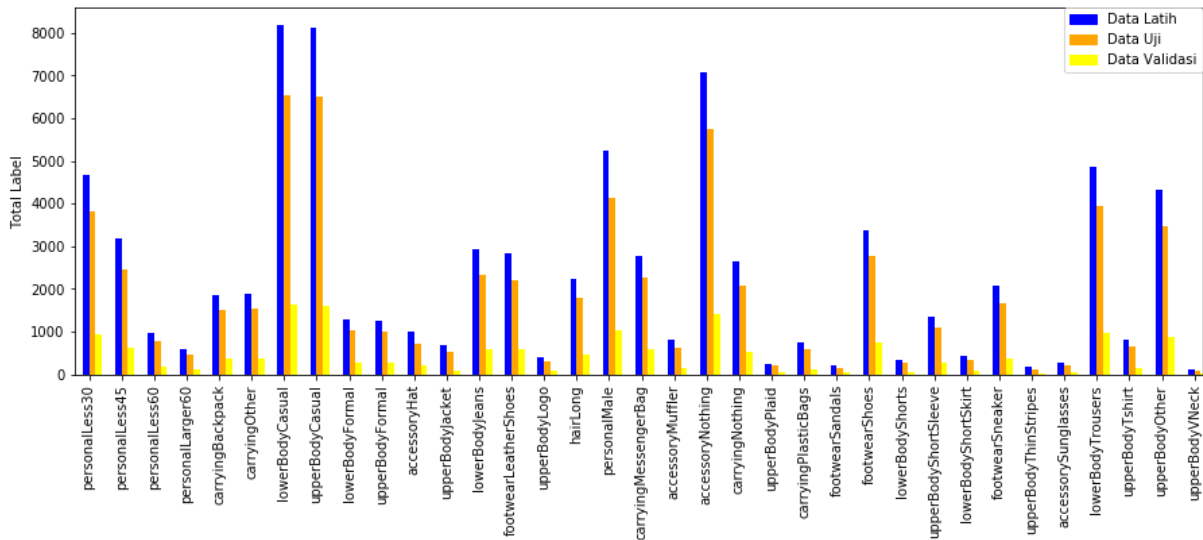
Gambar 6. Residual block

2.3 Pedestrian Attribute Dataset (PETA)



Gambar 7. Contoh beberapa gambar pada PETA

Dataset PETA disusun oleh Y. Deng dkk.[6] yang digunakan untuk penelitian pengenalan atribut pejalan kaki dengan metode SVM-Based. Dataset ini terdiri dari 19.000 gambar, dengan resolusi 17 x 39 hingga 169 x 365 piksel. Contoh beberapa gambar dapat dilihat di gambar 7. Atribut dari masing-masing gambar tersebut dianotasikan menjadi 105 label. Pada penelitian ini, pemilihan atribut yang digunakan dan persentase pembagian dataset juga mengacu pada penelitian yang dilakukan oleh Y.Deng dkk. yaitu sebanyak 35 label dan dengan pembagian dataset sebagai data latih sebesar 50% (9.500 gambar), data validasi 10% (1.900 gambar) dan data uji 40% (7.600 gambar). Berdasarkan pembagian tersebut, perbandingan jumlah dari masing-masing atribut terdapat pada Gambar 8.



Gambar 8. Pembagian Dataset

2.4 Pengukuran Kinerja

		Aktual	
		Positive	Negatif
Prediksi	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Tabel 1. Confusion Matrix

Pengukuran kinerja menggunakan *confusion matrix*. *Confusion matrix* merupakan metode yang sering digunakan untuk mengukur akurasi, *precision* dan *recall* dari suatu sistem klasifikasi. Terdapat 4 istilah untuk merepresentasikan hasil klasifikasi yaitu *True Positive* (TP), *True Negatif* (TN), *False Positive* (FP), dan *False Negative* (FN) seperti pada Tabel 1. Formulasi untuk menghitung akurasi, *precision* dan *recall* terdapat pada persamaan 1.

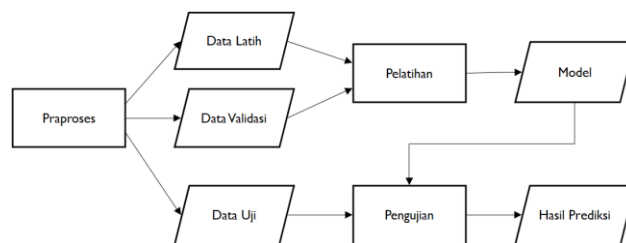
$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$$

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

....(1)

3. Sistem yang Dibangun



Gambar 9. Blok proses sistem

Diagram alir dari sistem yang dibangun dapat dilihat pada gambar 9. Pada penelitian ini dibangun sebuah model CNN secara *transfer learning* yang berasal dari arsitektur ResNet50. *Input layer* pada penelitian ini menyesuaikan dengan *input layer* pada ResNet50 yaitu 224x224 dan bobot awal jaringan pada penelitian ini juga menggunakan bobot dari ResNet50 yang telah dilatih dengan dataset ImageNet, namun pada *output layer/classifier* diganti dengan beberapa *layer* untuk menyesuaikan kasus pada penelitian ini.

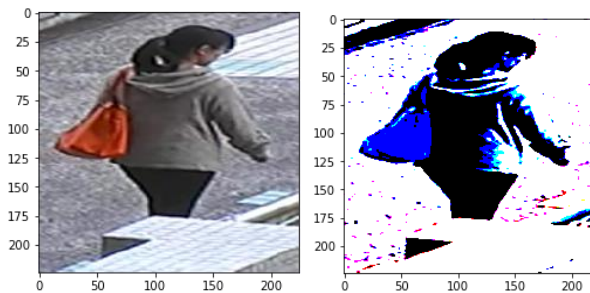
Setelah jaringan fitur ekstraksi pada ResNet50, kemudian disambungkan dengan satu *layer Global Average Pooling* (GAP) yang bertujuan untuk meringkas *feature map* kemudian disambungkan dengan dua layer yang berisi 1024 *neuron* dan 512 *neuron* yang masing-masing memiliki *dropout* 0,5 untuk mencegah *overfitting*, pada tahap terakhir dalam pembangunan arsitektur jaringan pada penelitian ini jaringan yang sudah ada disambungkan dengan *output layer* berisi 35 *neuron* sebagai layer klasifikasi. Layer terakhir pada penelitian ini menggunakan fungsi aktivasi sigmoid Arsitektur jaringan pada penelitian ini dapat dilihat pada tabel 9.

Layer	Keterangan
ResNet50	-
Global Average Pooling	-
Dense	Neuron : 1024, Fungsi aktivasi : ReLu
Dropout	0.5
Dense	Neuron : 512, Fungsi aktivasi : ReLu
Dropout	0.5
Dense	Neuron : 35, Fungsi aktivasi : Sigmoid

Tabel 2. Arsitektur Jaringan

3.1 Praproses

Penggunaan ukuran *input layer* dan bobot awal dari ResNet50 sehingga dilakukan *preprocessing* terlebih dahulu. Terdapat 2 *preprocessing* yang dilakukan yaitu *re-size* dan normalisasi piksel dataset. Ukuran gambar pada dataset PETA di-*re-size* keukuran 224x224 piksel dan normalisasi piksel sesuai karakteristik dari dataset ImageNet. Normalisasi dilakukan berdasarkan nilai *mean* dan standar deviasi dari dataset ImageNet. Normalisasi dilakukan dengan cara mengurangi setiap pixel dari dataset PETA [6] yang telah di-*re-size* dengan rata-rata nilai pixel dari dataset ImageNet kemudian dibagi dengan standar deviasi dari dataset ImageNet.



Gambar 10. Praproses

3.2 Pelatihan

Pada proses pelatihan menggunakan beberapa parameter yaitu optimizer adagrad, fungsi *loss binary crossentropy* dan *metric accuracy*. Pada proses pelatihan dilakukan validasi akurasi pada setiap *epoch* menggunakan data validasi. Monitoring dilakukan pada nilai *validation loss* untuk mengetahui kualitas pelatihan. Perubahan nilai *validation loss* juga diobservasi, observasi tersebut bertujuan untuk menyimpan bobot dan mengupdate *learning rate*. Jika terjadi penurunan nilai *loss validation*, maka bobot secara otomatis disimpan, namun jika terjadi peningkatan nilai *loss validation* dan atau tidak ada perubahan nilai *loss validation* dalam beberapa *epoch* berturut-turu maka *learning rate* akan otomatis diupdate dari *learning rate* sebelumnya. Pelatihan akan dihentikan ketika terjadi peningkatan *loss validation* yang secara signifikan yang menandakan telah terjadi *overfitting*.

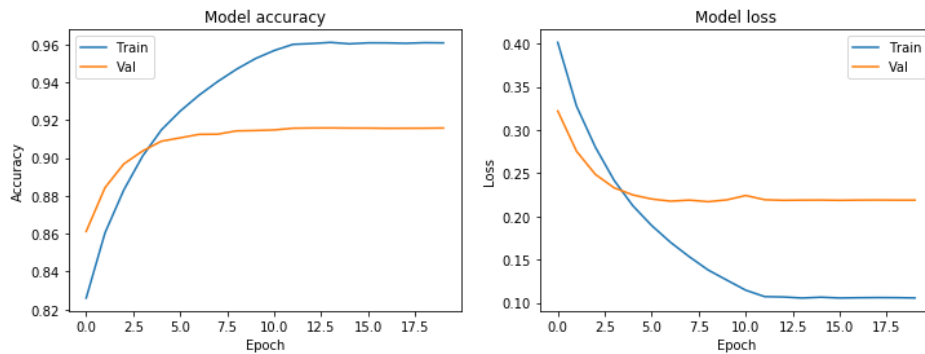
3.3 Pengujian

Output neuron merupakan hasil klasifikasi yang memiliki nilai antara 0 sampai 1, sehingga dibutuhkan *threshold* untuk menentukan label keluaran dari hasil klasifikasi. Penentuan *threshold* dilakukan dengan dua cara, pertama nilai *threshold* untuk semua *output neuron* bernilai sama yaitu 0.5 dan cara kedua adalah masing-masing *output neuron* memiliki nilai *threshold* berbeda-beda. Pada cara kedua, setiap neuron diobservasi dengan *threshold* antara 0.001 hingga 0.999 dengan penambahan 0.001 disetiap iterasi menggunakan data latih. Pada setiap iterasi dilakukan perhitungan akurasi, *precision*, dan *recall* untuk menentukan kualitas *threshold*, hingga didapatkan *threshold* untuk masing-masing *output neuron*. Pada proses pengujian menggunakan model dari hasil pelatihan dan nilai *threshold* dari cara pertama dan kedua yang diuji menggunakan data uji. Perhitungan akurasi, *precision* dan *recall* digunakan untuk mengetahui kinerja dari sistem yang dibangun.

3.4 Pengukuran Kinerja

Pada penelitian ini, pengukuran kinerja dilakukan untuk mengevaluasi sistem yang telah dibangun. Kinerja sistem dihitung berdasarkan TP, FP, TN, FN dari setiap gambar uji. Dari total jumlah TP, FP, TN dan FN digunakan untuk menghitung akurasi, *precision* dan *recall* [9].

4. Evaluasi



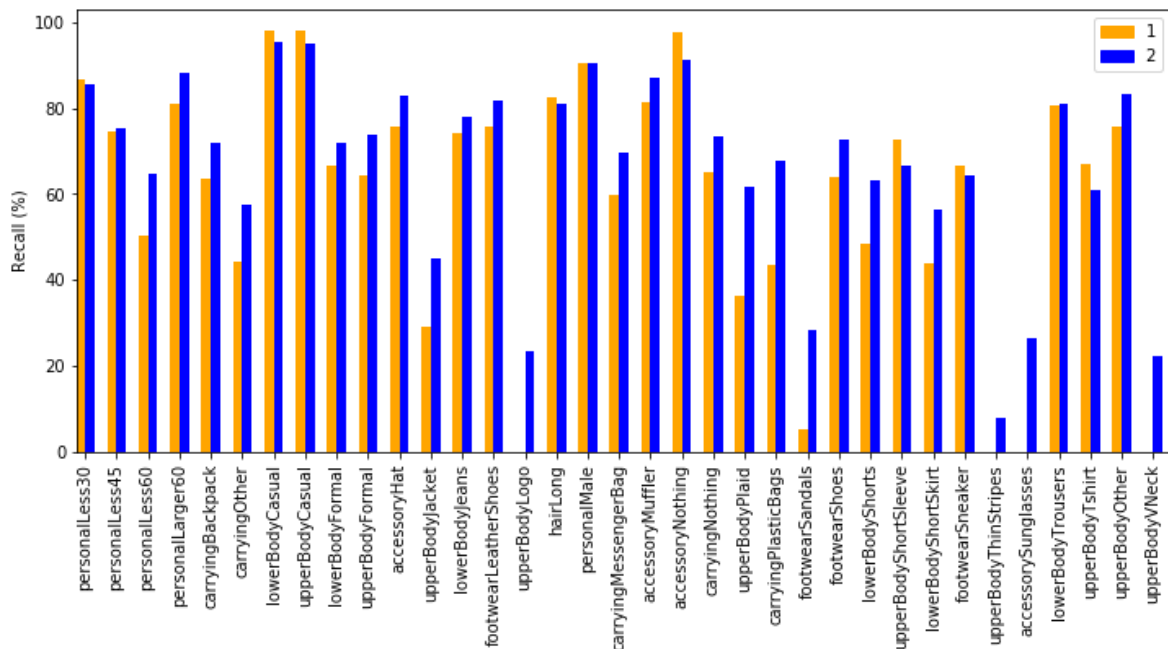
Gambar 11. Grafik pelatihan

Pada Gambar 11 adalah grafik dari hasil pelatihan yang dilakukan pada penelitian ini. Pelatihan dilakukan sebanyak 20 epoch. Namun mulai epoch ke 4, muncul tanda-tanda overfitting yang ditandai dengan grafik garis akurasi dan loss dari data validasi meningkat daripada data latih. Sehingga proses pelatihan hanya menyimpan model terbaik sebelum overfitting.

	Accuracy	Precision	Recall
Cara 1	91,56%	84,98%	78,78%
Cara 2	90.84%	80,83%	81,08%

Tabel 3. Hasil observasi threshold

Pada Tabel 3 ditampilkan hasil dari observasi yang telah dilakukan pada pencarian threshold terbaik menggunakan cara satu dan cara dua pada data validasi. Dari tabel tersebut, penggunaan threshold dengan nilai 0.5 pada seluruh output neuron menghasilkan akurasi dan *precision* yang lebih unggul. Kemudian cara kedua pada pencarian threshold untuk masing-masing neuron menghasilkan Recall yang lebih unggul dari pada cara kesatu.



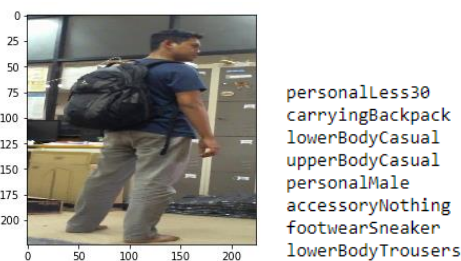
Gambar 12 Perbandingan hasil pencarian threshold terbaik

Untuk mengetahui kemampuan sistem dalam mengenali setiap atribut dapat dilihat dari *precision* atau *recall*. Pada Gambar 12 dapat dilihat bahwa dengan cara kesatu, sistem sama sekali tidak mampu mengenali adanya label *upperBodyLogo*, *upperBodyThinStripes*, *accessorySunglasses* dan *upperBodyVNeck*. Sehingga pada penelitian ini, threshold dari hasil observasi, pencarian threshold untuk masing-masing *output neuron* lebih baik dari pada semua neuron memiliki threshold yang sama (0,5) sehingga threshold cara kedua digunakan sebagai pengujian menggunakan data uji.

Label	MRFr2	DeepSAR	DeepMAR	Hasil Penelitian
personalLess30	86.8	82.9	85.8	84.78
personalLess45	83.1	79.4	81.8	82.76
personalLess60	80.1	83.3	86.3	93.34
personalLarger60	93.8	92.0	94.8	98.49
carryingBackpack	70.5	78.8	82.6	88.38
carryingOther	73.0	73.0	77.3	82.93
lowerBodyCasual	78.2	81.6	84.9	92.67
upperBodyCasual	78.1	81.1	84.4	92.22
lowerBodyFormal	79.0	81.9	85.2	92.97
upperBodyFormal	78.7	81.6	85.1	93.07
accessoryHat	90.4	89.2	91.8	96.28
upperBodyJacket	72.2	77.5	79.2	93.07
lowerBodyJeans	81.0	80.2	85.7	86.07
footwearLeatherShoes	87.2	84.2	87.3	87.91
upperBodyLogo	52.7	76.1	68.4	94.32
hairLong	80.1	83.2	88.9	91.08
personalMale	86.5	85.1	89.9	89.51
carryingMessengerBag	78.3	77.4	82.0	83.01
accessoryMuffler	93.7	94.4	96.1	97.83
accessoryNothing	82.7	81.5	85.8	87.34
carryingNothing	76.5	78.8	83.1	84.20
upperBodyPlaid	65.2	84.9	81.1	97.92
carryingPlasticBags	81.3	82.9	87.0	95.04
footwearSandals	52.2	81.3	67.3	97.05
footwearShoes	78.4	75.8	80.0	78.28
lowerBodyShorts	65.2	81.9	80.4	96.43
upperBodyShortSleeve	75.8	84.6	87.5	90.37
lowerBodyShortSkirt	69.6	83.2	82.2	96.64
footwearSneaker	75.0	77.3	78.7	85.08
upperBodyThinStripes	51.9	72.8	66.5	97.24
accessorySunglasses	53.5	79.1	69.9	96.55
lowerBodyTrousers	82.2	78.4	84.3	81.51
upperBodyTshirt	71.4	80.0	83.0	92.95
upperBodyOther	87.3	83.4	86.1	83.21
upperBodyVNeck	53.3	75.4	69.8	98.07
Rata-rata Akurasi	75.57	81.26	82.58	90.82

Tabel 4. Perbandingan hasil penelitian terhadap penelitian sebelumnya

Pada bagian ini ditampilkan hasil pengujian yang telah dilakukan menggunakan data uji pada model dan threshold yang dipilih. Sebagai perbandingan dengan penelitian sebelumnya, digunakan akurasi untuk mengetahui kinerja dari sistem yang dibangun. Berdasarkan rata-rata akurasi perkelas, penelitian ini mengalami peningkatan akurasi yang signifikan dari penelitian sebelumnya yang menggunakan ekstraksi fitur konvensional para MRFr2. Peningkatan juga terjadi dibandingkan dengan beberapa penelitian yang juga menggunakan arsitektur CNN [9][10][11][12] Peningkatan ini dapat dilihat pada tabel 4. Pada gambar 13 merupakan hasil uji menggunakan foto di luar dataset. Hasilnya sudah sesuai namun pada bagian kaki terdeteksi menggunakan *sneaker* padahal kaki pada objek tersebut tidak menggunakan apapun, namun karena perbedaan warna yang cukup mencolok pada kaki sehingga dikenali sebagai *sneaker*.



Gambar 13. Contoh hasil uji menggunakan gambar diluar dataset

5. Kesimpulan

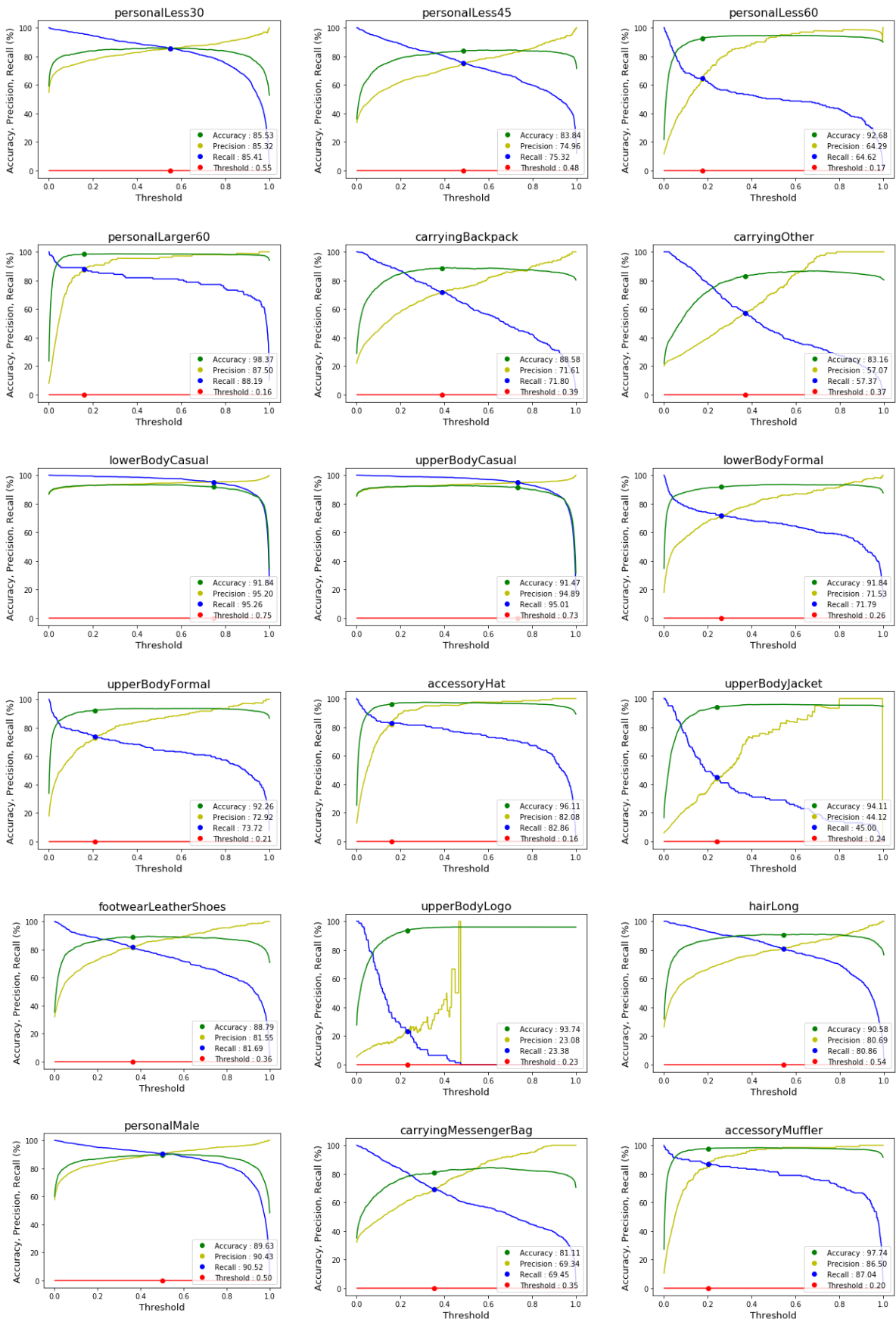
Berdasarkan penelitian ini, dapat ditarik kesimpulan bahwa dalam menentukan threshold dapat dilakukan dengan pencarian dalam rentang nilai threshold tertentu untuk setiap output neuron sehingga dapat diketahui nilai threshold untuk setiap neuron yang mampu menghasilkan nilai akurasi/precision/recall yang lebih baik. Dengan menggunakan data uji yang sama persis dengan penelitian yang sudah ada, hasil dari akurasi rata-rata pada penelitian ini yaitu 90,82%, hal ini mengalami peningkatan yang signifikan daripada penelitian sebelumnya menggunakan fitur ekstraksi konvensional maupun dengan metode CNN. Peningkatan tersebut dapat dilihat pada tabel 4.

Pada penelitian ini, observasi hyperparameter sangat minim dilakukan. Meskipun demikian mampu mencapai peningkatan akurasi rata-rata yang signifikan. Diharapkan pada pengembangan selanjutnya dapat dilakukan observasi hyperparameter yang mendalam agar didapatkan parameter yang lebih optimal sehingga mampu meningkatkan akurasi saat ini.

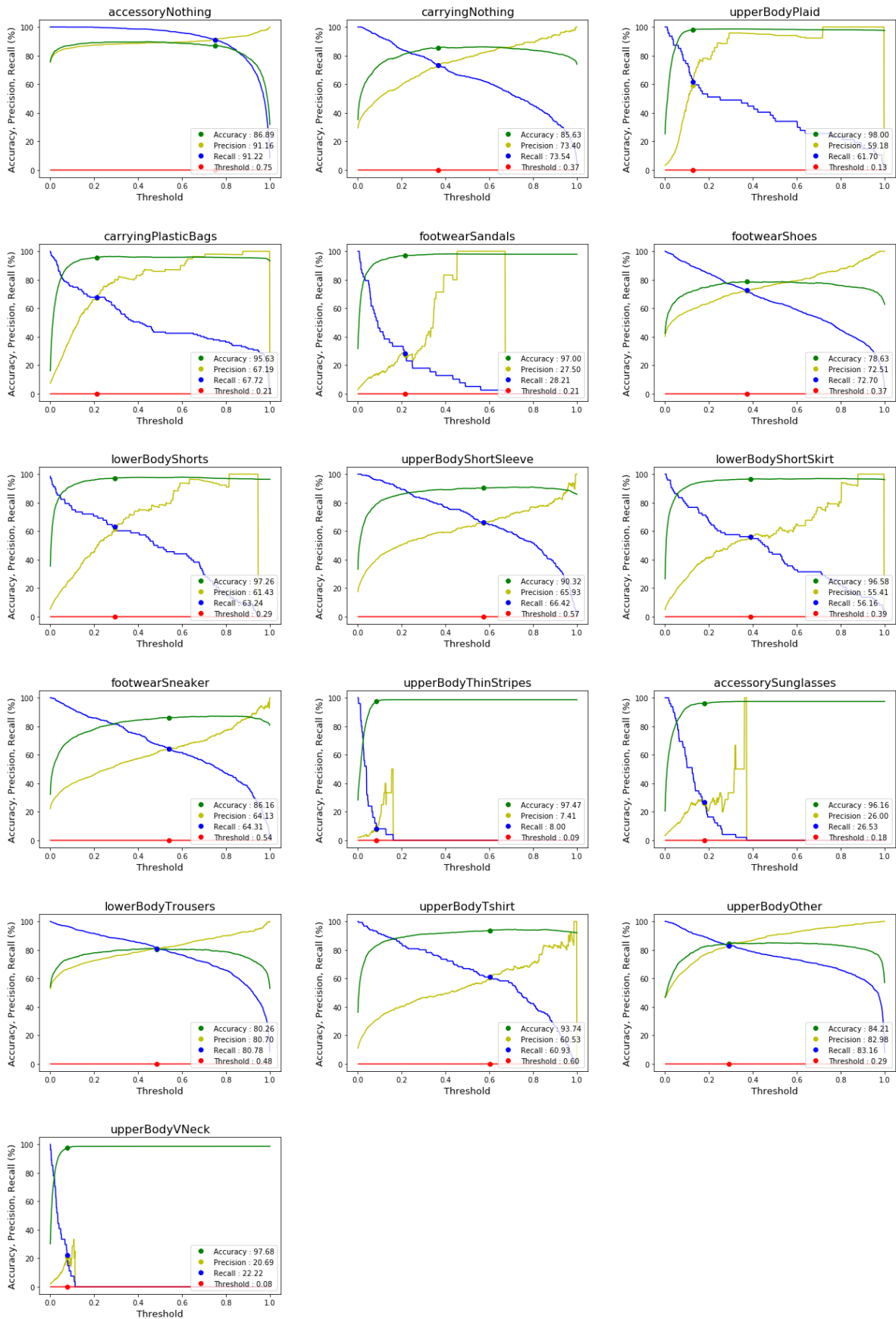
Daftar Pustaka

- [1] R. Layne, T. Hospedales, and S. Gong, "Person Re-identification by Attributes," *Proceedings Br. Mach. Vis. Conf. 2012*, no. January, pp. 24.1-24.11, 2012, doi: 10.5244/C.26.24.
- [2] Y. Deng, P. Luo, C. C. Loy, and X. Tang, "Learning to Recognize Pedestrian Attribute," no. February, 2015.
- [3] D. Ganda and R. Buch, "A Survey on Multi Label Classification," *Recent Trends Program. Lang.*, vol. 5, no. August, 2018.
- [4] D. Li, X. Chen, and K. Huang, "Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios," *Proc. - 3rd IAPR Asian Conf. Pattern Recognition, ACPR 2015*, no. November, pp. 111–115, 2016, doi: 10.1109/ACPR.2015.7486476.
- [5] C. C. Chang and C. J. Lin, "LIBSVM: A Library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1–39, 2011, doi: 10.1145/1961189.1961199.
- [6] Y. DENG, P. Luo, C. C. Loy, and X. Tang, "Pedestrian Attribute Recognition At Far Distance," pp. 789–792, 2014, doi: 10.1145/2647868.2654966.
- [7] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: A review," *Neurocomputing*, vol. 187, pp. 27–48, 2016, doi: 10.1016/j.neucom.2015.09.116.
- [8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," 2015.
- [9] X. Wang, S. Zheng, R. Yang, B. Luo, and J. Tang, "Pedestrian Attribute Recognition: A Survey," vol. 14, no. 8, pp. 1–32, 2019.
- [10] X. Liu *et al.*, "HydraPlus-Net: Attentive Deep Features for Pedestrian Analysis," *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2017-Octob, no. c, pp. 350–359, 2017, doi: 10.1109/ICCV.2017.46.
- [11] A. Gupta and J. Ramesh, "Pedestrian Attribute Detection using CNN," *csn231Report*, p. 6, 2016.
- [12] Z. Ji, W. Zheng, and Y. Pang, "Deep pedestrian attribute recognition based on LSTM," *Proc. - Int. Conf. Image Process. ICIP*, vol. 2017-Sept, pp. 151–155, 2018, doi: 10.1109/ICIP.2017.8296261.

Lampiran



Gambar. Hasil pencarian threshold setiap label



Gambar. Hasil pencarian threshold setiap label