Mini Report Face Detection

Febrian Abdul Fatah, Alfachran Caesar, Aina Indah Lestari, Tasya Nurfitria

1. **Pendahuluan**

Computer Face Detection atau yang disebut pendeteksi wajah merupakan topik yang menjadi pusat perhatian akhir-akhir ini. Terlebih pada satu decade terakhir menimbulkan peningkatan signifikan dalam ketertarikan pada *face detection* baik dalam pusat studi penelitian maupun jumlah personil yang terlibat dalam hal penerapan, metode ataupun Teknik yang diusulkan pada *face detection* ini.

Penggunaan pendeteksi wajah saat ini telah banyak digunakan oleh berbagai kalangan dalam berbagai bidang, seperti aplikasi telepon pintar, CCTV, TV pintar dll. Namun pada pengaplikasiannya banyak diantara hasil pendeteksi wajah tidak akurat dalam mendeteksi wajah seseorang. Pengertian pendeteksi wajah menurut *Wikipedia* yang dikutip dari www.facedetection.com ialah teknologi komputer yang digunakan dalam berbagai aplikasi untuk mengidentifikasi wajah manusia menggunakan gambar digital[2]. Beberapa metode yang kerap digunakan oleh beberapa peneliti diantaranya adalah Neural Network, OpenCV, Deep Learning dan lain sebagainya.

Terdapat beberapa hal yang membuat face detection menjadi pusat perhatian dalam pemrosesan komputer. Pertama, terdapat factor ekonomi yang kuat didalamnya:   
Algoritma pengolah dan pengenalan wajah yang dapat diandalkan akan memberikan implikasi pada sector komersial. Diantaranya sebagai berikut [3]

Keamanan : Dapat digunakan untuk mengidentifikasi seseorang yang berwenang untuk dapat mengakses area tertentu.

Pelayanan (Akses informasi dan layanan) : Mengidentifikasi pemilik Kartu kredit, mengakses computer dan jaringan, memverfikasi pengguna untuk akses ke pelayanan kesehatan dlsb.

Penegak Hukum : Kontrol paspor, mengidentifikasi seseorang terhadap database, membuat sketsa dari deskripsi para saksi.

Terdapat bidang lain yang berpotensi salah satunya seperti pengobatan forensik (rekonstruksi wajah atau identifikasi orang dan sisa-sisa kerangka), konfrensi video dll. Pada mini paper ini kami menyajikan beberapa hasil untuk pengenalan wajah manusia menggunakan jaringan saraf *Multi Layer Perceptron* dan *Logistic Regression*.

**II. Data Engineering**

Dataset yang digunakan ialah berupa foto-foto *selfie* dengan kriteria usia mulai dari 3– 60 tahun, berjenis kelamin laki-laki dan perempuan. Dataset yang digunakan ialah foto-foto *selfie* yang memiliki ekstensi .png (re:dot png) file, yang dikumpulkan dari semua Mahasiswa Ilmu Komputer angkatan 2016 berjumlah 267 foto dan 160 foto *selfie* selain Mahasiswa Ilmu Komputer angkatan 2016. Dari seluruh dataset yang tersedia, 320 foto kami gunakan sebagai dataset yang akan dieksperimen, terdiri dari 20 mahasiswa yang masing-masingnya memiliki 8 foto dengan ekspresi berbeda dan 20 orang selain mahasiswa ilmu komputer 2016 yang masing-masing juga memiliki 8 foto *selfie* berbeda.

Kemudian data tersebut diolah dengan Photoshop CS6 untuk diubah ukuran dimensinya menjadi 56x56 pixel dari ukuran yang sebenarnya. Area yang kami gunakan dari dataset foto tersebut adalah bagian wajah, kerudung dan rambut, sehingga kami menghapus background dan objek lainnya di dalam foto tersebut. Untuk sebagian foto yang wanita kami bedakan yaitu ada yang diambil hanya bagian wajahnya dan ada juga yang diambil beserta kerudung/rambut. Kemudian kami save data yang sudah diolah tersebut dalam bentuk .png file.

[1] A. Samal and P. A. Iyengar, “Automatic Recognition and Analysis of Human Faces and Facial Expressions: A Survey. Pattern Recognition”, Vol. 25, No. 1, pp 65-72, 1992.

[2] <https://facedetection.com> (Diakses pada 4 Mei 2018)

[3] R. Chellappa, C. L. Wilson and S. Sirohey, “Human and MachineRecognition of Faces: A Survey”, Proceedings of the IEEE. Vol. 83. No. 5, pp 705-740, May 1995.



**Gambar 1.** Contoh dari 8 perbedaan eskpresi *selfie* pada pria

**III. Neural Network Configuration**

Percobaan yang dilakukan meliputi beberapa *neuron* yang dimodifikasi dan di uji. Pada kasus ini, setiap *neuron* pada *hidden layer* diuji, dari satu sampai empat *neuron*. Adapula *input layer* dan *output layer* pada percobaan, yang masing-masing bernilai 12.544 (didapatkan dari dataset berdimensi 3 yang digunakan yang berukuran 56x56x4) dan 40. Dari beberapa teori yang dicakupkan, ukuran *hidden layer* yang optimal biasanya berada diantara ukuran *input* dan *output layer*.

Pada tahap percobaan, Multilayer Perceptron yang digunakan dikonfigurasikan dengan *value* yang sama, yakni:

* learning\_rate = 0.01
* batch\_size = 512
* n\_epoch = 90
* n\_train = 240
* n\_test = 80

Yang kemudian dilanjut dengan modifikasi *value* daripada Multilayer Perceptron. Percobaan yang dilakukan mencakup uji coba setiap *neuron*, sehingga didapat akurasi yang tepat.

**Experiment**

Berikut percobaan berserta hasil akurasi uji coba yang didapatkan.

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment** | **Neuron Size** | **Accuracy** | **Time** |
| 1 | 1000 | 0.125 | 10.133 seconds |
| 2 | 700 | 0.2 | 8.366 seconds |
| 3 | 500 | 0.175 | 6.895 seconds |
| 4 | 200 | **0.3** | 5.178 seconds |
| 5 | 120 | 0.05 | 5.147 seconds |
| 6 | 70 | 0.125 | 4.544 seconds |
| 7 | 45 | 0.2 | 4.426 seconds |

**Tabel 1.** Percobaan dengan 1 *Neuron*

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment** | **Neuron Size** | **Accuracy** | **Time** |
| 1 | 1000, 700 | 0.0375 | 10.492 seconds |
| 2 | 1000, 100 | 0.4 | 10.396 seconds |
| 3 | 700, 350 | 0.125 | 8.544 seconds |
| 4 | 700, 45 | 0.025 | 8.275 seconds |
| 5 | 250, 200 | 0.025 | 5.714 seconds |
| 6 | 250, 50 | 0.3875 | 5.674 seconds |
| 7 | 70, 45 | **0.4125** | 4.594 seconds |

**Tabel 2.** Percobaan dengan 2 *Neuron*

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment** | **Neuron Size** | **Accuracy** | **Time** |
| 1 | 1000, 800, 200 | 0.1375 | 10.630 seconds |
| 2 | 1000, 100, 50 | 0.375 | 10.256 seconds |
| 3 | 700, 600, 500 | 0.025 | 8.764 seconds |
| 4 | 700, 70, 45 | 0.225 | 8.505 seconds |
| 5 | 550, 120, 70 | 0.3625 | 7.775 seconds |
| 6 | 550, 70, 45 | **0.4** | 7.678 seconds |
| 7 | 100, 70, 50 | 0.3625 | 4.892 seconds |

**Tabel 3.** Percobaan dengan 3 *Neuron*

|  |  |  |  |
| --- | --- | --- | --- |
| **Experiment** | **Neuron Size** | **Accuracy** | **Time** |
| 1 | 1000, 800, 200, 100 | 0.1625 | 10.987 seconds |
| 2 | 1000, 100, 60, 45 | 0.0875 | 10.997 seconds |
| 3 | 800, 300, 200, 80 | 0.1875 | 22.373 seconds |
| 4 | 800, 80, 60, 45 | 0.1375 | 11.208 seconds |
| 5 | 400, 300, 200, 100 | **0.25** | 6.908 seconds |
| 6 | 400, 120, 80, 50 | 0.1375 | 6.723 seconds |
| 7 | 110, 80, 60, 45 | 0.1 | 6.692 seconds |

**Tabel 4.** Percobaan dengan 4 *Neuron*

Berdasarkan penelitian yang telah dilakukan untuk tiap-tiap *neuron*, hasil akurasi yang paling tinggi adalah **0.4125** yang terdiri atas dua *neuron* pada *hidden layer*. Keluaran yang dihasilkan masih terbilang kurang akurat. Dengan beberapa percobaan lagi, serta penambahan jumlah *epoch*, akurasi yang dihasilkan dapat lebih optimal.

**IV.** **Eksperimen dan Evaluasi**

Untuk membandingkan hasil yang didapatkan dari Multi Layer Perceptron, Metode Regresi logistik diterapkan sebagai pembanding hasil. Regresi Logistik adalah sebuah pendekatan untuk membuat model prediksi seperti halnya Regresi Linear. Namun regresi logistik digunakan untuk memprediksi keluaran (*outcome*) dari variabel yang hanya dapat mengambil *value* yang spesifik dan terbatas (*categorical variable*). Singkatnya, pada Regresi Logistik peneliti memprediksi variabel terikat yang berskala dikotomi. Skala dikotomi yang dimaksud adalah skala data nominal dengan dua kategori.

Pada tahap percobaan, Logistic Regression yang digunakan dikonfigurasikan dengan 2 *value* yang berbeda, yakni:

Value 1 :

* learning\_rate = 0.01
* batch\_size = 512
* n\_epoch = 10-130
* n\_train = 240
* n\_test = 80

Value 2 :

* learning\_rate = 0.01
* batch\_size = 128
* n\_epoch = 10-120
* n\_train = 240
* n\_test = 80

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Experimen | Batch | Epochs | Train | Test | Accuracy | Total Time |
| 1 | 512 | 10 | 240 | 80 | 0,1875 | 0.7830426692962646 |
| 2 | 512 | 20 | 240 | 80 | 0,1375 | 1.3986492156982422 |
| 3 | 512 | 30 | 240 | 80 | 0,3125 | 1.5731053352355957 |
| 4 | 512 | 40 | 240 | 80 | 0,325 | 2.0624804496765137 |
| 5 | 512 | 50 | 240 | 80 | 0,3625 | 1.999387264251709 |
| 6 | 512 | 60 | 240 | 80 | 0,475 | 2.8454225063323975 |
| 7 | 512 | 70 | 240 | 80 | 0,4375 | 3.209371566772461 |
| 8 | 512 | 80 | 240 | 80 | 0,5 | 3.7116830348968506 |
| 9 | 512 | 90 | 240 | 80 | 0,475 | 3.953258752822876 |
| 10 | 512 | 100 | 240 | 80 | 0.4625 | 4.6828906536102295 |
| 11 | 512 | 110 | 240 | 80 | 0,5 | 4.74237060546875 |
| 12 | 512 | 120 | 240 | 80 | 0,5375 | 5.065212965011597 |
| 13 | 512 | 130 | 240 | 80 | 0,5 | 5.910418272018433 |

**Tabel 7.** Batch 512 Logistic Regression

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Experimen | Batch | Epochs | Train | Test | Accuracy | Total Time |
| 1 | 128 | 10 | 240 | 80 | 0,2 | 0.8442020416259766 |
| 2 | 128 | 20 | 240 | 80 | 0,3375 | 1.356619119644165 |
| 3 | 128 | 30 | 240 | 80 | 0,5 | 1.7725937366485596 |
| 4 | 128 | 40 | 240 | 80 | 0,45 | 2.2583324909210205 |
| 5 | 128 | 50 | 240 | 80 | 0.525 | 2.73714017868042 |
| 6 | 128 | 60 | 240 | 80 | 0,475 | 3.1682655811309814 |
| 7 | 128 | 70 | 240 | 80 | 0,5375 | 3.618438720703125 |
| 8 | 128 | 80 | 240 | 80 | 0,4625 | 4.107297897338867 |
| 9 | 128 | 90 | 240 | 80 | 0,5125 | 4.920198678970337 |
| 10 | 128 | 100 | 240 | 80 | 0,5125 | 4.939883708953857 |
| 11 | 128 | 110 | 240 | 80 | 0,45 | 5.441190958023071 |
| 12 | 128 | 120 | 240 | 80 | 0,4875 | 5.8201820850372314 |

Berdasarkan tabel 1 dan tabel 2, Epoch berfungsi untuk melakukan update terhadap nilai variable/parameter sehingga fungsi menuju akurasi yang optimal. Penelitian ini dilakukan dengan 25 experiment yang bertujuan untuk menunjukan akurasi yang paling optimal melalui 25 experiment tersebut.

**Tabel 8.** Batch 128 Regresi Logistik

Penelitian ini mengambil 2 batch yang berbeda pada tabel 1 batch yang diambil adalah 512 dan pada tabel 2 batch yang diambil adalah 128 , dan adanya perbedaan batch ini bertujuan untuk membuat perbandingan untuk mencari akurasi yang optimal dan yang terbaik dalam dataset ini dengan menggunakan Logistic Regression.

Pada tabel 1 dengan batch 512 akurasi yang optimal dan yang terbaiknya tertuju pada experiment ke-12 epoch 120 dengan nilai akurasinya 0,5375 dan dengan waktu 5.910418272018433 second. Pada tabel 2 dengan batch 128 akurasi yang optimal dan yang terbaiknya tertuju pada experiment ke-20 epoch 70 dengan nilai akurasinya 0,5375 dan dengan waktu 3.618438720703125 second.

**V. Kesimpulan**

Face Detection pada percobaan ini diujikan melalui dua metode yaitu Neural Network Multi Layer Perceptron dan Logistic regression dengan konfigurasi yang berbeda-beda. Dengan menggunakan dataset yang dikumpulkan melalui metode sampel dan di lakukan pengaturan pada background didapat hasil yang optimal untuk mengecek wajah seseorang.

Hasil eksperimen juga menunjukan penggunaan metode Neural Network Multi Layer Perceptron memiliki akurasi 0,4125 dengan jumlah 3 neuron, dengan masing-masing 550,70, dan 45 dengan total waktu yang dibutuhkan 7,678 second waktu maksimal, dibanding dengan Metode Regresi Logistik dengan jumlah dataset yang sama menghasilkan nilai akurasi yang lebih tinggi yaitu 0,5375 dengan epoch 120 dan waktu 5,91 detik. Perlu dilakukan tindakan yang lebih konferhensif terkait penelitian dan percobaan ini dengan jumlah neuron yang berbeda beda pada multi layer perceptron dan kajian yang lebih rinci terhadap *face detection.*