

# Pengenalan Angka Sistem Isyarat Bahasa Indonesia Dengan Menggunakan Metode Convolutional Neural Network

Mochamad Bagus Setiyo Bakti<sup>1</sup>, Yuliana Melita Pranoto<sup>2</sup>

<sup>1,2</sup>Teknologi Informasi, Sekolah Tinggi Teknik Surabaya

E-mail: <sup>1</sup>bagussetyabakti@gmail.com, <sup>2</sup>ymp@stts.edu

**Abstrak** –Bahasa isyarat adalah metode komunikasi dengan menggunakan gerakan tangan yang biasanya dilakukan oleh tuna rungu. Di Indonesia sendiri mempunyai 2 jenis bahasa isyarat yaitu BISINDO dan SIBI (Sistem Isyarat Bahasa Indonesia). Perbedaannya dalam isyarat dasar BISINDO menggunakan dua tangan sedangkan SIBI hanya satu tangan. Dalam paper ini akan membahas pengenalan isyarat angka SIBI dengan menggunakan metode Convolutional Neural Network (CNN). Arsitektur CNN yang digunakan adalah arsitektur LeNet. Arsitektur CNN diproses dalam 3 tahap, 25 epoch, 50 epoch dan 100 epoch. Berdasarkan percobaan yang dilakukan nilai akurasi yang didapat terus meningkat dalam tiap tahapnya, mulai 67.66%, 89.44% sampai nilai akurasi tertinggi dalam proses training sebesar 96.44%. Begitupun dalam proses prediksi data juga mengalami kenaikan dalam tiap tahapnya, mulai 79.23%, 90.45% sampai didapatkan nilai akurasi tertinggi dalam prediksi data 98.89%. Dari total 90 data testingset, hanya sekali keasalahan dalam prediksi data.

**Kata Kunci** — SIBI, CNN, LeNet, Machine Learning

## 1. PENDAHULUAN

Indonesia memiliki total penduduk sekitar 264 juta jiwa. Indonesi menempati urutan ketiga dengan jumlah penduduk terbesar di Asia setelah China dan Inda. Dalam hal disabilitas khususnya tunarungu, Indonesia menempati peringkat keempat setelah Srilanka, Myanmar dan India dengan total 16.8% penduduk atau setara 35 juta jiwa. Diperkirakan setiap tahunnya ada lebih dari 5 ribu bayi terlahir dalam tuli.

Dalam kehidupan sosial tuna rungu sulit dalam berkomunikasi dengan orang lain. Tuli atau tuna rungu adalah hilangnya kemampuan untuk mendengar dari salah satu atau kedua telinga. Tuna rungu cenderung mempunyai alat bantu pendengaran atau menggunakan bahasa isyarat dalam berkomunikasi. Kebanyakan orang menggunakan gerakan tangan atau tubuh dalam melakukan suatu komunikasi. Gerakan tersebut yang kemudian dinamakan bahasa isyarat.

Bahasa isyarat adalah metode komunikasi yang tidak menggunakan suara, tetapi menggunakan gerakan tangan, tubuh dan bibir untuk menyampaikan sebuah informasi. Bahasa isyarat telah digunakan sebagai sistem komunikasi, terutama oleh penderita tuli atau tuna rungu. [1]. Bahasa isyarat di setiap Negara memiliki tanda yang berbeda. *American Sign Language* (ASL) [2] adalah salah satu bahasa visual yang lengkap dengan menggunakan tanda-tanda yang dibuat melalui gerakan tangan yang dikombinasikan dengan ekspresi wajah juga gestur tubuh. ASL terdiri dari 6000 gerakan. Ejaan jari digunakan untuk berkomunikasi dengan kata-kata yang sulit dipahami. Ejaan jari terdiri dari 26 gerakan tangan yang menunjukkan 26 alfabet.

Selain itu, di India bahkan mendirikan sebuah sekolah khusus untuk penyandang cacat, tuli dan buta. The Mathru Educational School memberikan pendidikan gratis kepada para penyandang cacat, tuli serta buta. Sekolah ini memiliki pendidik khusus bahasa isyarat yang *reliable*. Di Mathru, guru menjelaskan objek atau konsep melalui gambar. Karena para guru memiliki kosakata tanda isyarat tertentu untuk menjelaskan, mereka sering menunjukkannya melalui gambar tangan yang ada di poster.[3]

Di Indonesia mempunyai 2 Bahasa Isyarat yang digunakan, Bahasa Isyarat Indonesia (BISINDO) [4] dan Sistem Isyarat Bahasa Indonesia (SIBI). BISINDO merupakan penyesuaian dari *American Sign Language* (ASL) dengan budaya asli Indonesia sehingga mudah digunakan. BISINDO menggunakan gerakan 2 tangan untuk berinteraksi. Sedangkan SIBI merupakan bahasa isyarat yang mengadopsi ASL. SIBI secara resmi digunakan oleh semua SLB dibawah Kementerian Pendidikan dan Kebudayaan (Kemendikbud). Peraturan penggunaan SIBI sudah diatur dalam peraturan Menteri Pendidikan Republik Indonesia nomor 0161/U/1994 [5]. SIBI memiliki 26 ejaan jari yang menunjukkan 26 alfabet dengan menggunakan satu tangan. 24 tanda berupa gerakan statis serta 2 tanda berupa gerakan dinamis (J dan Z).

Penelitian tentang bahasa isyarat banyak dilakukan diseluruh negara. Dampak sosial yang dihasilkan banyak dirasakan para penyandang tuna rungu. Tujuan dalam penelitian ini adalah untuk mengenalkan dasar bahasa isyarat untuk pembelajaran. Sehingga para penderita tuna rungu dapat berkomunikasi dengan orang normal tanpa terbatas oleh penyampaian bahasa. Kebanyakan penelitian menggunakan metode *computer*

vision. Di era teknologi 4.0 *machine learning* menjadi salah satu metode yang sangat populer. Pada dasarnya *machine learning* adalah proses dimana komputer belajar beberapa hal yang belum pernah dikenal dengan sumber dari banyak data. Dalam pemrosesan gambar, *Convolutional Neural Network (CNN)* adalah salah satu algoritma yang sangat digemari. Dalam penelitian ini CNN digunakan untuk mentraining data sehingga bisa mengenali data berupa angka SIBI. Dengan menggunakan CNN diharapkan nilai akurasi yang didapatkan bisa lebih tinggi dengan nilai loss yang kecil.

## 2. METODE PENELITIAN

Untuk dapat memprediksi keluaran berupa angka SIBI maka akan dilakukan 2 proses. Yang pertama *preprocessing gambar* untuk mendapatkan area tangan yang akan digunakan untuk proses *training* dan *testing*. Yang kedua adalah proses *training* dan *testing* data dengan CNN. Untuk penjelasan lengkap terkait prosesnya akan dijelaskan pada setiap subbabnya.

### 2.1 Daftar Kelas

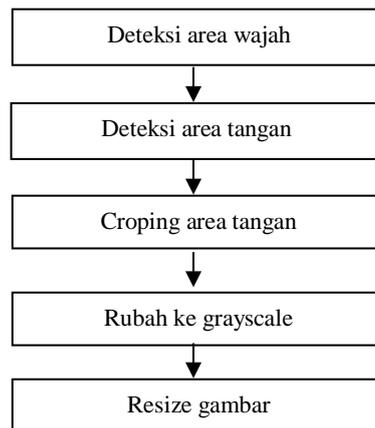
Dataset yang disediakan berupa foto peragaan alphabet SIBI yang diperagakan oleh model. Model difoto dengan menggunakan kamera *smartphone*. Setiap peraga dilakukan pengambilan data sebanyak 3 kali untuk setiap peragaan. Total dataset yang didapat sebanyak 450 gambar. Daftar kelas keluaran yang dihasilkan sebanyak 10 kelas angka mulai dari 0 – 9. Contoh bentuk alfabet SIBI beserta artinya seperti pada table 1.

Tabel 1. Daftar Kelas Angka SIBI

Isyarat	Angka	Isyarat	Angka
	0		5
	1		6
	2		7
	3		8
	4		9

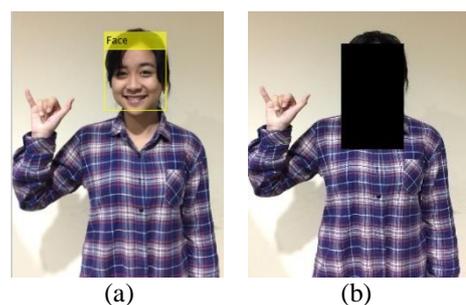
### 2.2 Preprocessing

Tahap *preprocessing* dilakukan dengan tujuan untuk mendapatkan area tangan agar mempermudah proses *training CNN*. Tahapan *preprocessing* yang dilakukan antara lain deteksi area wajah, deteksi warna kulit, *cropping* area tangan serta merubah warna gambar ke *grayscale*. Alur *preprocessing* dapat dilihat seperti pada gambar 1.



Gambar 1. Proses *Preprocessing*

Langkah pertama yang dilakukan dalam tahap *preprocessing* adalah melakukan deteksi wajah. Deteksi area wajah menggunakan algoritma *viola jones* [6]. Setelah area wajah ditemukan, selanjutnya dilakukan pengurangan citra di area wajah. Area leher juga dihilangkan dengan cara menambahkan setengah kali area wajah ke arah bawah, sehingga area wajah dan leher dapat dihilangkan. Proses tersebut dilakukan untuk memudahkan proses *hand detection*.



Gambar 2. (a) Hasil deteksi wajah (b) hasil menghapus area wajah dan leher

Setelah area wajah dan leher dihilangkan, langkah selanjutnya melakukan deteksi warna kulit [7]. Pada umumnya tiap orang mempunyai warna kulit yang berbeda. Untuk memudahkan dalam mengenali warna kulit, ruang warna RGB pada gambar ditransformasikan ke ruang warna Ycbr. Transformsinya seperti pada matrix 1.

$$\begin{bmatrix} Y \\ C_r \\ C_b \end{bmatrix} = \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.5 & -0.419 & -0.018 \\ -0.169 & -0.331 & 0.5 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

Dari hasil matrix diatas diperoleh kisaran nilai dasar warna kulit [8] dimana nilai  $C_r$ ,  $C_b$  dan  $Y$  adalah [133, 173], [77, 127], [0,255]. Setelah area tangan ditemukan, langkah selanjutnya adalah menghapus seluruh area gambar kecuali area tangan yang didapatkan sebelumnya. Hasilnya seperti pada gambar 3.



Gambar 3. Hasil deteksi warna kulit

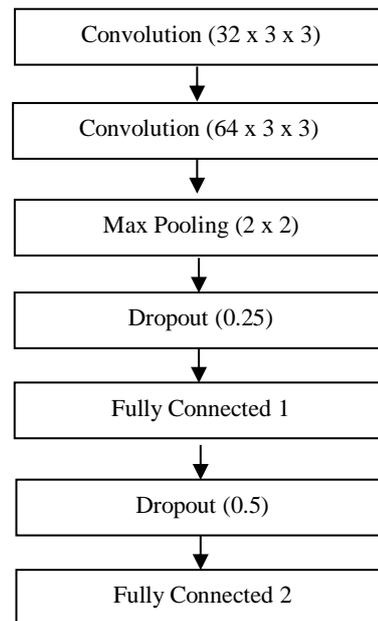
Setelah diperoleh area tangan, langkah selanjutnya adalah melakukan *cropping gambar* pada area tersebut. Selanjutnya gambar hasil *cropping* di *resize* ke ukuran 64x64 px. Proses terakhir dari *preprocessing* adalah merubah warna gambar menjadi *grayscale* dengan tujuan agar gambar mudah diproses oleh CNN dan nilai akurasi yang didapat lebih besar. Hasilnya seperti pada tabel 2.

Tabel 2. Hasil Preprocessing

Gambar	Angka	Gambar	Angka
	0		5
	1		6
	2		7
	3		8
	4		9

### 2.3 Arsitektur CNN

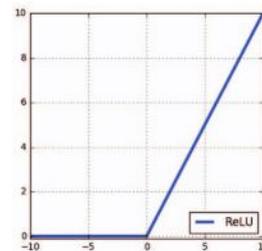
Dalam penelitian ini menggunakan struktur CNN LeNet yang pernah digunakan dalam penelitian Yiliang Xie, et al [9]. Gambar 4 menunjukkan arsitektur CNN yang digunakan. Ukuran input gambar 64 x 64 x 1. Ruang warna yang digunakan adalah *grayscale*. *Convolutional layer* 1 memiliki kernel ukuran 3x3 sebanyak 32. Setelah itu diikuti *convolutional layer* yang kedua dengan 64 layer ukuran 3x3. Selanjutnya diikuti *max pooling* ukuran 2x2. Ditambahkan pula *regularization dropout* untuk menghindari *overfitting*. Layer terakhir ditutup dengan *fully connected layer* 10 keluaran untuk mengelompokkan 10 kelas angka.



Gambar 4. Struktur CNN

Dalam setiap *convolution layer* terdapat *hidden layer* dengan *activation function ReLu*. ReLu bisa membantu proses komputasi menjadi lebih cepat dibandingkan tanh [10]. ReLu juga dapat membantu untuk mengurangi *overfitting*. Pada rumus 2 menjelaskan tentang fungsi reLu.

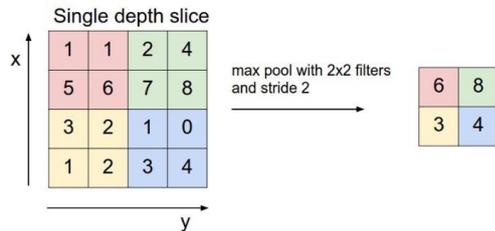
$$f(x) = \max(0, x) \dots \dots \dots (2)$$



Gambar 5. ReLu Activation

*Pooling layer* berisi sebuah filter dengan ukuran 2x2 yang prosesnya akan bergeser keseluruhan bagian dari

*feature map*. Dalam penelitian ini menggunakan *max pooling*. *Max pooling* mengambil nilai tertinggi dari area yang dilewati. Tujuannya adalah untuk mengurangi dimensi gambar sehingga dapat mempercepat proses komputasi serta dapat mengatasi permasalahan *overfitting*.



Gambar 6. Proses *max pooling*

Nilai *dropout* diatur 0.25 sebelum *fully connected layer* pertama dan 0.5 sebelum *fully connected layer* kedua. *Learning algorithm* yang digunakan adalah *ADADELTA* [11].

Setelah dilakukan proses training, akan didapatkan nilai akurasi serta nilai *loss* dari training data yang dilakukan. Selain itu dalam proses training juga didapatkan *weight model*. Model tersebut dapat disimpan dan *doload* kembali untuk melakukan proses prediksi data.

### 3. HASIL DAN PEMBAHASAN

Proses training menggunakan PC core i5, memory 8 GB serta GPU 8 GB. GPU digunakan untuk melakukan training dan testing data *machine learning*. Dalam melakukan proses training dan testing menggunakan aplikasi *python*. Library yang digunakan adalah *Tensorflow* dan *Keras*. Library tersebut merupakan library yang sangat baik untuk memproses *machine learning*.

Sebelum dilakukan training, dataset dibagi menjadi 2 bagian yaitu data training dan data testing yang digunakan untuk prediksi. Dari total 450 data dibagi 80% untuk data training dan 20% untuk data testing. Proses training dan testing data dilakukan sebanyak 3 tahapan, mulai dari training sebanyak 25 *epoch*, 50 *epoch* dan 100 *epoch*. Hal tersebut dilakukan dengan tujuan untuk melihat nilai akurasi pada setiap tahap.

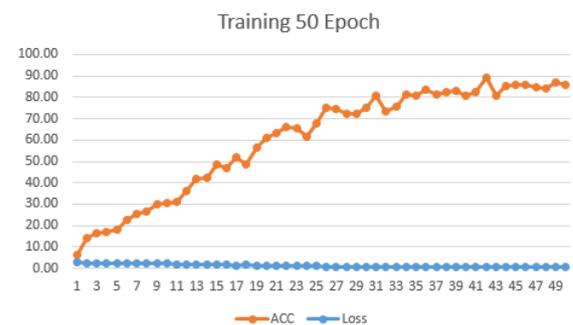
Tahap pertama dilakukan training sebanyak 25 *epoch*. Hasilnya nilai akurasi yang didapat sebesar 67.66% dengan nilai *loss* 0.95%. Nilai akurasi tersebut dirasa masih kurang baik. Selanjutnya dilakukan proses prediksi terhadap 20% data testing yang sudah disiapkan. Hasilnya nilai akurasi yang didapatkan saat proses prediksi sebesar 79.23%. Dari total 90 gambar prediksi, ada kesalahan prediksi sebanyak 19 gambar.

Dari total 10 kelas keluaran hanya angka 0 yang mampu memprediksi hingga 100%.



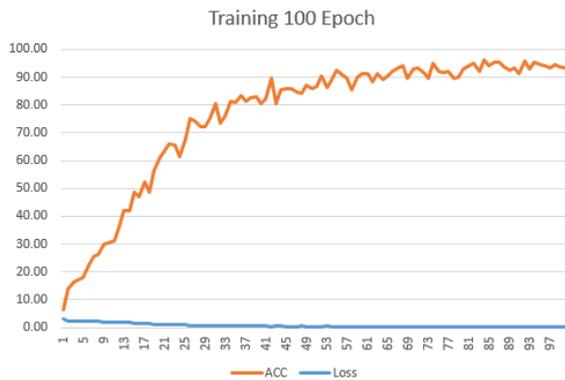
Gambar 7. Nilai akurasi & *loss* training 25 *epoch*

Tahap kedua dataset dilakukan training kembali sebanyak 50 *epoch*. Hasilnya nilai akurasinya meningkat menjadi 89.44% dengan nilai *loss* yang juga menurun menjadi 0.37%. Nilai akurasi yang didapat pada tahap ini meningkat dari sebelumnya yang hanya 67.66%. Dari model yang sudah ditraining sebelumnya, dilakukan proses prediksi pada data testing. Hasilnya nilai akurasi prediksi meningkat menjadi 90.44%. Total *error* prediksi juga menurun dari sebelumnya 19 gambar menjadi 9 gambar.



Gambar 8. Nilai akurasi & *loss* training 50 *epoch*

Tahap ke tiga dilakukan proses training kembali sebanyak 100 *epoch*. Hasilnya nilai akurasi mencapai 96.44% dengan nilai *loss* yang juga turun menjadi 0.13%. Selanjutnya pengujian dilakukan dengan data testing. Hasilnya nilai akurasi prediksi data juga ikut meningkat menjadi 98.89%. Dalam tahap ini kesalahan prediksi data hanya terjadi 1 kali dari total 90 gambar.



Gambar 9. Nilai akurasi & loss training 100 epoch

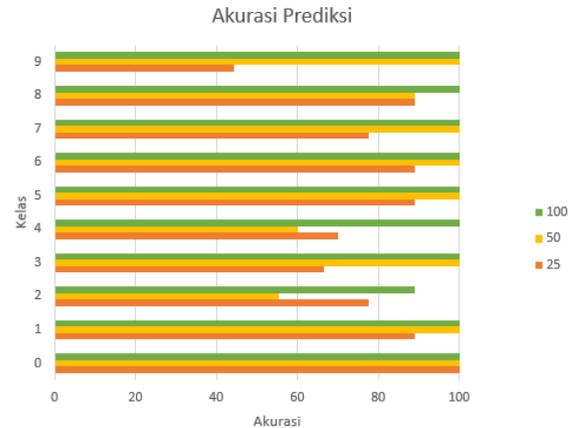
Hasil dari pemrosesan dataset baik training maupun testing mengalami kenaikan pada setiap epochnya. Dalam pemrosesan data testing nilai akurasi yang didapat dalam setiap tahap selalu mengalami kenaikan dari 67.66%, 89.44%, dan terakhir meningkat menjadi 96.44%. Begitupun dengan nilai loss yang turun pada setiap epochnya. Hal ini menunjukkan bahwa semakin banyak epoch yang digunakan dalam training data, kemungkinan nilai akurasi yang didapat semakin baik.

Sedangkan dalam proses prediksi data juga mengalami kenaikan akurasi. Dalam tahap pertama akurasi yang didapat 79.23% dengan kesalahan prediksi 19 gambar. Hampir pada semua kelas mengalami kesalahan dalam prediksi data, hanya pada kelas angka 0 yang dapat diprediksi 100% benar. Pada tahap kedua nilai akurasi meningkat menjadi 90.45% dengan kesalahan prediksi menurun menjadi 9 gambar. Kesalahan prediksi hanya terjadi pada 3 kelas saja yaitu kelas angka 2 dan 7. Pada tahap terakhir nilai akurasi testing kembali meningkat menjadi 98.89% dengan 1 kali kesalahan prediksi data. Kesalahan prediksi data hanya terjadi pada 1 kelas dan 1 gambar saja yaitu kelas angka 2. Untuk mengetahui detail akurasi saat proses prediksi data dapat dilihat pada tabel 3.

Tabel 3. Hasil prediksi data tes

Kelas	25 Epoch		50 Epoch		100 Epoch	
	Akurasi (%)	Error	Akurasi (%)	Error	Akurasi (%)	Error
0	100	-	100	-	100	-
1	88.89	2	100	-	100	-
2	77.78	7,9	55.56	7,9,8,9	88.89	9
3	66.67	5,2,2	100	0	100	-
4	70	5,5,5	60	5,5,5,5	100	-
5	88.89	7	100	-	100	-
6	88.89	9	100	-	100	-

7	77.78	5,4	100	-	100	-
8	88.89	7	88.89	7	100	-
9	44.44	4,2,3,2,8	100	-	100	-



Gambar 10. Grafik akurasi prediksi tiap kelas

#### 4. SIMPULAN

Dalam era modern saat ini mempelajari bahasa isyarat sangat mudah karena banyaknya media yang memfasilitasi hal tersebut. Hal ini tentu bisa memberikan dampak yang baik dalam kehidupan sosial. *Machine learning* menjadi salah satu metode pembelajaran yang paling populer saat ini. Metode *Convolutional Neural Network (CNN)* dapat menjadi salah satu metode untuk memecahkan masalah bahasa isyarat dengan hasil keluaran yang sangat baik.

Struktur CNN dapat dimodifikasi sesuai kebutuhan. *LeNet* menjadi salah satu struktur CNN yang sesuai dalam pemrosesan gambar. Terbukti dengan menggunakan arsitektur *LeNet* tingkat akurasi training dan testing mencapai 90% bahkan lebih. Dalam proses training setiap tahapnya dapat mempengaruhi nilai akurasi. Terbukti dari 25 sampai 100 epoch yang telah dilakukan, nilai akurasi dalam training data terus meningkat hingga mencapai 96.44% dengan nilai loss 0.13%. Hal tersebut juga dibuktikan dalam hasil keluaran dari proses testing data. Data test dapat diprediksi dengan baik hingga akurasinya mencapai 98.89%. Dari percobaan yang telah dilakukan, proses prediksi data pada kelas angka 2 selalu mengalami kesalahan prediksi data.

#### 5. SARAN

Tentunya masih banyak kekurangan dalam penelitian ini. Kedepan akan ada pengembangan dalam penelitian selanjutnya. Ada banyak harapan terutama bagi mereka yang berkebutuhan khusus untuk bisa merasakan hal yang sama dengan mereka yang normal. Maka akan banyak penelitian dalam bidang sosial yang akan diselesaikan menggunakan *machine learning*.

DAFTAR PUSTAKA

- [1] Sulfayanti, Dewiani, and Armin Lawi, "A real time alphabets sign language recognition system using hands tracking," *2016 Int. Conf. Comput. Intell. Cybern.*, pp. 69–72, 2016.
- [2] C. M. Jin, Z. Omar, and M. H. Jaward, "A mobile application of American sign language translation via gambar processing algorithms," *2016 IEEE Reg. 10 Symp.*, pp. 104–109, 2016.
- [3] T. Dharamsi, R. Jawahar, K. Mahesh, and G. Srinivasa, "Stringing Subtitles in Sign Language," *2016 IEEE Eighth Int. Conf. Technol. Educ.*, pp. 228–231, 2016.
- [4] I. Kautsar, R. I. Borman, and A. Sulistyawati, "Aplikasi pembelajaran bahasa isyarat bagi penyandang tuna rungu berbasis android dengan metode bisindo," *Semin. Nas. Teknol. Inf. dan Multimed. 2015*, vol. 3, no. 2003, pp. 6–8, 2015.
- [5] Menteri Pendidikan dan Kebudayaan Republik Indonesia "Pembakuan Sistem Isyarat Bahasa Indonesia Bagi Kaum Tuna Rungu" No 0161/U/1994, 1994.
- [6] A. Cordiner, "Illumination invariant face detection," *MComSc thesis, Univ. Wollongong*, p. 175, 2009.
- [7] S. Yang and Q. Zhu, "Video-based Chinese sign language recognition using convolutional neural network," *2017 9th IEEE Int. Conf. Commun. Softw. Networks, ICCSN 2017*, vol. 2017–January, pp. 929–934, 2017.
- [8] D. Chai and K. N. Ngan, "Locating facial region of a head-and-shoulders color gambar," *Proc. - 3rd IEEE Int. Conf. Autom. Face Gesture Recognition, FG 1998*, pp. 124–129, 1998.
- [9] Y. Xie, H. Jin, and E. C. C. Tsang, "Improving the lenet with batch normalization and online hard example mining for digits recognition," *Int. Conf. Wavelet Anal. Pattern Recognit.*, vol. 1, pp. 149–153, 2017.
- [10] G. Antonellis, A. G. Gavras, M. Panagiotou, B. L. Kutter, G. Guerrini, A. C. Sander, and P. J. Fox, "Shake Table Test of Large-Scale Bridge Columns Supported on Rocking Shallow Foundations," *J. Geotech. Geoenvironmental Eng.*, vol. 12, p. 04015009, 2015.
- [11] Matthew D. Zeiller " Adadelta : An Adaptive Learning Rate Mehod" Cornell University, 2012.
- [12] CNN Indonesia, "Masih Ada 40,5 Juta Orang Indonesia Berjuang Lawan Ketulian", 2017. available : <https://www.cnnindonesia.com/gaya-hidup/20170323180138-255-202335/masih-ada-405-juta-orang-indonesia-berjuang-lawan-ketulian>, diakses 25 Agustus 2018.