

# **Laporan Tugas Final Deep Learning**

## **Semester Genap 2020/2021**

### **Klasifikasi Alphabet ISL dengan EfficientNet**



#### **Anggota Kelompok:**

Tomy Widjaja	- C14180006
Enrico Hartanto	- C14180013
Henry Wicaksono	- C14180014
Andrew Firman	- C14180029
Gregorius Nicholas	- C14180046
Gregorius Kevin	- C14180057

## I. Introduction

*Sign language* merupakan bahasa yang digunakan untuk berkomunikasi dengan orang yang tuna rungu. Untuk berkomunikasi menggunakan *sign language*, diperlukan keterampilan dalam mengartikan *sign language* tersebut. Oleh karena itu, diperlukan sistem yang dapat mengklasifikasikan gambar bentuk tangan untuk *Indonesian Sign Language*. Dengan demikian, teknologi *Deep Learning* dapat dimanfaatkan untuk pembelajaran *Indonesian Sign Language* secara otomatis.

## II. Related Works

- A. Sign Language Interpreter using Deep Learning (Gupta et al., 2019)
  1. Metode  
Deep Learning, CNN Model.
  2. Dataset  
*American Sign Language gestures*.
  3. Hasil  
Mampu mengklasifikasi sign language dengan akurasi lebih besar dari 96%.
- B. Pengenalan Bahasa Isyarat Indonesia (BISINDO) Untuk Karakter Huruf Dengan Menggunakan Microsoft Kinect (Yunanda et al., 2018)
  1. Metode  
Hidden Markov Model (HMM)
  2. Dataset  
Gerakan tangan yang diambil dengan menggunakan *Microsoft Kinect* meliputi:
    - Huruf: a, i, u, e, o.
    - Kata: berita, kunjung, rambut
  3. Hasil  
Uji coba dilakukan dengan pengambilan gerakan tangan dengan jarak 80 cm dan 150 cm dari sensor. Hasilnya adalah akurasi antara 75% untuk jarak 150 cm dan 87,5% untuk jarak 80 cm.

### III. Dataset

Dataset yang gunakan dalam proyek ini terdiri atas gambar berupa foto tangan yang membentuk seluruh alfabet *Indonesian Sign Language* (ISL) dan diambil dari dua sumber:

- Dataset BISINDO  
URL: <https://www.kaggle.com/idhamozi/indonesian-sign-language-bisindo>.
- Dataset hasil foto sendiri



Gambar 3.1 Contoh Dataset

#### A. Dataset Pengujian 1:

- Training Data
  - 64 foto tangan dari dataset BISINDO
  - **Total:** 1728 foto (64 foto per kelas)
- Testing Data
  - 2 foto tangan Henry per kelas
  - **Total:** 54 foto (2 foto per kelas)

#### B. Dataset Pengujian 2 (Dataset Sendiri)

- Training Data
  - 10 foto tangan Andrew per kelas
  - 10 foto tangan Enrico per kelas

- 10 foto tangan Gregorius per kelas
  - 10 foto tangan Henry per kelas
  - 10 foto tangan Nicholas per kelas
  - 10 foto tangan dari dataset BISINDO per kelas
  - **Total:** 1620 foto (60 foto per kelas)
- **Seen Testing Data** (Tangan terdapat pada training data)
  - 2 foto tangan Andrew per kelas
  - 2 foto tangan Enrico per kelas
  - 2 foto tangan Gregorius per kelas
  - 2 foto tangan Henry per kelas
  - 2 foto tangan Nicholas per kelas
  - 2 foto tangan dari dataset BISINDO per kelas
  - **Total:** 324 foto (12 foto per kelas)
- **Unseen Testing Data** (Tangan tidak terdapat pada training data)
  - 15 foto tangan Tomy per kelas
  - **Total:** 405 foto (15 foto per kelas)

#### IV. Method

Implementasi dilakukan dalam bahasa pemrograman Python dengan menggunakan library Tensorflow dan Keras. Proses transfer learning menggunakan EfficientNet B0 dengan parameter sebagai berikut:

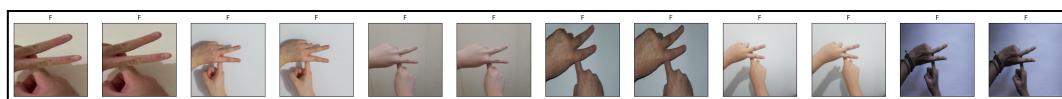
- Loss function: Categorical Cross-Entropy
- Optimizer: Rectified Adam
  - Learning rate = 1e-3
  - min\_lr = 1e-7
  - warmup\_proportion = 0.15
- Callback:
  - Learning Rate Reducer
    - monitor = 'val\_accuracy'
    - patience = 5
    - verbose = 1
    - factor = 0.2
    - min\_lr = 1e-7
  - Early Stopping
    - monitor = 'val\_accuracy'

- mode = 'max'
- patience = 5
- restore\_best\_weights = True

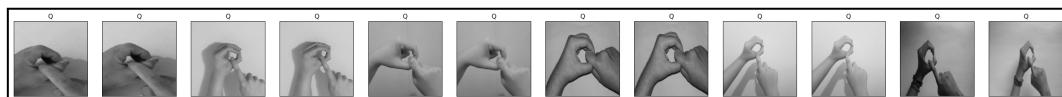
## V. Experiments

Seluruh image yang digunakan dalam pengujian akan melalui proses resize terlebih dahulu untuk mengubah dimensinya menjadi 224x224x3. Pengujian juga dilakukan dengan beberapa metode preprocessing yang berbeda:

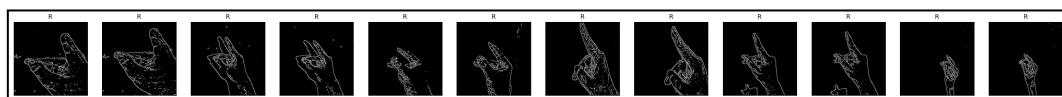
- Raw (Tanpa preprocessing)
- Grayscale
- Edge Detection
- Segmentation
- Sharpening



**Gambar 5.1 Raw Data (Tanpa Preprocessing)**



**Gambar 5.2 Grayscale**



**Gambar 5.3 Edge Detection**



**Gambar 5.4 Segmentation**



**Gambar 5.5 Sharpening**

Pengujian pertama menggunakan dataset BISINDO yang terdiri atas 64 image per kelasnya sebagai training data. Model hasil training kemudian diujikan pada data testing yang merupakan hasil foto sendiri. Data testing terdiri atas 2 image per kelasnya. Hasil pengujian dapat dilihat pada tabel 6.1.

Setelah melalui berbagai tahapan preprocessing dan percobaan, didapati bahwa hasil akurasi model tetap tidak memuaskan pada pengujian pertama. Hal ini diperkirakan disebabkan oleh dataset training yang hanya terdiri dari tangan satu orang yang sama, meskipun jumlah image per labelnya cukup banyak. Model pun menjadi overfit dan hanya bisa mengenali tangan orang pada data training, sehingga akurasinya sangat buruk saat digunakan untuk mengklasifikasikan testing data yang merupakan hasil foto buatan sendiri. Oleh karena itu, dibuatlah dataset sendiri yang terdiri atas tangan 6 orang berbeda agar dataset training lebih bervariasi.

Pengujian kedua dilakukan dengan menggunakan dataset buatan sendiri (selanjutnya disebut sebagai Dataset Sendiri). Pada pengujian ditemukan bahwa model dengan konfigurasi unfreeze top 20 layers pada EfficientNet B0 menghasilkan akurasi yang lebih baik daripada konfigurasi unfreeze all layers. Konfigurasi unfreeze top 20 layers ini yang kemudian akan digunakan dalam pengujian perbandingan berbagai metode preprocessing pada pengujian kedua:

- Dataset Sendiri - Raw (Tanpa Preprocessing) (Tabel 6.2)
- Dataset Sendiri - Grayscale (Tabel 6.3)
- Dataset Sendiri - Edge Detection (Tabel 6.4)
- Dataset Sendiri - Segmentation (Tabel 6.5)
- Dataset Sendiri - Sharpening (Tabel 6.6)

Perbandingan setiap metode preprocessing secara keseluruhan pada pengujian kedua kemudian dipaparkan pada tabel 6.7

## VI. Results

Pengujian model yang ditraining pada dataset BISINDO menunjukkan hasil yang kurang memuaskan dengan nilai F1-Score 0.35. Hal ini disebabkan karena dataset BISINDO hanya terdiri atas tangan orang yang sama meskipun jumlah foto per kelasnya banyak. Hal ini menyebabkan model tidak akurat saat digunakan untuk memprediksi foto tangan orang lain. Hasil pengujian dapat dilihat pada tabel 6.1.

**Tabel 6.1 Dataset BISINDO - Unseen Test Data**

Class	Precision	Recall	F1-Score
Overall	0.35	0.35	0.35
Nothing	1.0	1.0	1.0
A	0.0	0	0

B	0.0	0	0
C	0.0	0	0
D	0.0	0	0
E	1.0	0.5	0.67
F	0.0	0	0
G	0.0	0	0
H	0.0	0	0
I	0.0	0.0	0
J	0.0	0	0
K	0.0	0	0
L	1.0	0.29	0.44
M	1.0	0.33	0.5
N	0.0	0	0
O	1.0	0.2	0.33
P	0.0	0	0
Q	0.0	0	0
R	0.5	1.0	0.67
S	1.0	0.67	0.8
T	0.5	1.0	0.67
U	1.0	0.29	0.44
V	0.0	0.0	0
W	0.0	0	0
X	1.0	1.0	1.0

Y	0.0	0	0
Z	0.5	0.11	0.18

Pengujian dengan Dataset Sendiri menunjukkan hasil yang jauh lebih baik dibandingkan pengujian pertama. Hal ini dikarenakan data training terdiri atas tangan orang yang berbeda-beda. Dengan demikian, model lebih robust dalam mengenali bermacam-macam foto tangan. Pengujian tanpa preprocessing menunjukkan nilai F1-Score 0.91 untuk Seen Test Data dan 0.81 untuk Unseen Test Data. Namun, terdapat kasus di mana ada kelas yang memiliki F1-Score yang baik pada Seen Data namun sama sekali gagal diprediksi pada Unseen Data, seperti huruf H dan U. Hasil pengujian untuk raw dataset dapat dilihat pada tabel 6.2.

**Tabel 6.2 Dataset Sendiri - Raw**

Class	Seen Test Data			Unseen Test Data		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Overall	0.91	0.91	0.91	0.81	0.81	0.81
Nothing	1.0	1.0	1.0	1.0	1.0	1.0
A	1.0	1.0	1.0	1.0	0.94	0.97
B	1.0	1.0	1.0	1.0	1.0	1.0
C	0.83	0.77	0.8	1.0	0.43	0.6
D	1.0	0.67	0.8	0.87	0.81	0.84
E	1.0	1.0	1.0	0.6	1.0	0.75
F	0.83	1.0	0.91	0.67	1.0	0.8
G	1.0	1.0	1.0	1.0	1.0	1.0
H	0.92	1.0	0.96	0.0	0	0
I	0.92	1.0	0.96	0.33	1.0	0.5
J	1.0	1.0	1.0	1.0	1.0	1.0

K	0.75	1.0	0.86	0.4	0.86	0.55
L	0.83	1.0	0.91	1.0	0.71	0.83
M	0.92	0.69	0.79	1.0	0.94	0.97
N	0.42	0.83	0.56	0.93	1.0	0.97
O	1.0	0.92	0.96	1.0	0.83	0.91
P	0.5	1.0	0.67	0.87	0.81	0.84
Q	1.0	1.0	1.0	1.0	1.0	1.0
R	1.0	1.0	1.0	1.0	0.94	0.97
S	1.0	0.92	0.96	0.4	1.0	0.57
T	1.0	0.8	0.89	0.93	0.4	0.56
U	0.75	0.82	0.78	0.0	0.0	0.0
V	0.92	1.0	0.96	0.93	0.61	0.74
W	1.0	1.0	1.0	1.0	0.94	0.97
X	1.0	0.86	0.92	1.0	1.0	1.0
Y	1.0	0.71	0.83	1.0	0.75	0.86
Z	1.0	1.0	1.0	1.0	0.94	0.97

Pengujian dengan preprocessing grayscale justru menghasilkan akurasi yang lebih rendah dibandingkan raw dataset. F1-Score menunjukkan angka 0.9 untuk Seen Data dan 0.7 untuk Unseen Data. Hal ini kemungkinan besar disebabkan akibat warna kulit dari beberapa foto tangan yang hampir mirip dengan warna background foto, sehingga proses grayscaling justru membuat model kesulitan dalam mengekstrak fitur dengan tepat. Hasil pengujian untuk preprocessing grayscale dapat dilihat pada tabel 6.3.

**Tabel 6.3 Dataset Sendiri - Grayscale**

Class	Seen Test Data			Unseen Test Data		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
<b>Overall</b>	<b>0.9</b>	<b>0.9</b>	<b>0.9</b>	<b>0.7</b>	<b>0.7</b>	<b>0.7</b>
Nothing	1.0	1.0	1.0	1.0	1.0	1.0
A	1.0	1.0	1.0	1.0	1.0	1.0
B	1.0	1.0	1.0	0.47	0.7	0.56
C	0.75	0.64	0.69	1.0	0.54	0.7
D	1.0	0.92	0.96	0.53	1.0	0.7
E	1.0	0.8	0.89	1.0	1.0	1.0
F	0.83	0.71	0.77	0.47	0.88	0.61
G	1.0	1.0	1.0	0.87	1.0	0.93
H	0.83	1.0	0.91	0.33	1.0	0.5
I	0.83	1.0	0.91	0.2	0.75	0.32
J	0.92	1.0	0.96	0.6	1.0	0.75
K	1.0	1.0	1.0	0.87	0.5	0.63
L	1.0	0.75	0.86	0.93	0.38	0.54
M	0.67	1.0	0.8	0.13	1.0	0.24
N	0.83	0.71	0.77	1.0	0.54	0.7
O	1.0	1.0	1.0	1.0	0.62	0.77
P	0.83	1.0	0.91	1.0	0.42	0.59
Q	1.0	1.0	1.0	0.47	1.0	0.64
R	1.0	0.75	0.86	0.87	0.87	0.87

S	0.75	1.0	0.86	0.2	0.33	0.25
T	1.0	0.86	0.92	0.67	0.56	0.61
U	0.75	0.75	0.75	0.33	0.62	0.43
V	0.58	1.0	0.74	0.6	1.0	0.75
W	0.83	1.0	0.91	0.33	0.83	0.48
X	1.0	0.86	0.92	1.0	1.0	1.0
Y	0.83	1.0	0.91	1.0	0.79	0.88
Z	1.0	0.92	0.96	0.93	0.88	0.9

Pengujian untuk preprocessing berupa edge detection justru menghasilkan akurasi terendah. F1-Score hanya menunjukkan nilai 0.27 untuk Seen Data dan 0.12 untuk Unseen Data. Hal ini kemungkinan besar disebabkan oleh banyaknya sparse data (pixel dengan value 0). Bahkan, 18 dari 26 kelas gagal diprediksi sama sekali untuk Unseen Data. Hasil pengujian edge detection dapat dilihat pada tabel 6.4.

**Tabel 6.4 Dataset Sendiri - Edge Detection**

Class	Seen Test Data			Unseen Test Data		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Overall	0.27	0.27	0.27	0.12	0.12	0.12
Nothing	1.0	1.0	1.0	0.6	1.0	0.75
A	0.33	1.0	0.5	0.0	0.0	0.0
B	0.58	0.12	0.21	0.0	0.0	0.0
C	0.0	0.0	0.0	0.0	0.0	0.0
D	0.0	0.0	0.0	0.0	0.0	0.0
E	0.08	0.12	0.1	0.0	0.0	0.0
F	0.42	0.2	0.27	0.0	0.0	0.0

G	0.0	0.0	0.0	0.0	0.0	0.0
H	0.0	0.0	0.0	0.0	0.0	0.0
I	0.33	0.67	0.44	0.27	0.5	0.35
J	0.0	0.0	0.0	0.0	0.0	0.0
K	0.08	0.25	0.12	0.0	0.0	0.0
L	0.42	0.62	0.5	0.27	0.29	0.28
M	0.25	0.6	0.35	0.0	0.0	0.0
N	0.92	0.15	0.27	1.0	0.06	0.11
O	0.67	0.67	0.67	0.4	0.75	0.52
P	0.08	0.2	0.12	0.0	0.0	0.0
Q	0.0	0.0	0.0	0.0	0.0	0.0
R	0.17	0.08	0.11	0.07	0.5	0.12
S	0.5	0.26	0.34	0.07	0.06	0.06
T	0.0	0.0	0.0	0.0	0.0	0.0
U	0.0	0.0	0.0	0.0	0.0	0.0
V	0.75	0.3	0.43	0.33	0.2	0.25
W	0.0	0.0	0.0	0.0	0.0	0.0
X	0.67	0.62	0.64	0.2	1.0	0.33
Y	0.0	0.0	0.0	0.0	0.0	0.0
Z	0.17	0.14	0.15	0.0	0.0	0.0

Pengujian pada preprocessing segmentation menghasilkan nilai F1-Score yang cukup baik pada Seen Data, yakni 0.84. Namun, pengujian pada Unseen Data menghasilkan nilai F1-Score yang sangat rendah, yakni hanya 0.49. Hal ini menunjukkan bahwa segmentation rentan terhadap overfitting dan kurang baik

dalam memprediksi tangan yang belum pernah dilihatnya sebelumnya. Hasil pengujian untuk segmentation dapat dilihat pada tabel 6.5.

**Tabel 6.5 Dataset Sendiri - Segmentation**

Class	Seen Test Data			Unseen Test Data		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
<b>Overall</b>	<b>0.84</b>	<b>0.84</b>	<b>0.84</b>	<b>0.49</b>	<b>0.49</b>	<b>0.49</b>
Nothing	0.92	1.0	0.96	0.73	0.79	0.76
A	1.0	1.0	1.0	0.8	1.0	0.89
B	0.92	1.0	0.96	0.93	0.93	0.93
C	1.0	0.86	0.92	0.13	0.4	0.2
D	0.75	1.0	0.86	0.0	0.0	0
E	0.92	0.92	0.92	0.73	0.55	0.63
F	0.83	0.77	0.8	0.53	0.4	0.46
G	0.58	1.0	0.74	0.4	0.46	0.43
H	0.92	1.0	0.96	0.27	0.18	0.22
I	0.92	0.73	0.81	0.27	0.25	0.26
J	1.0	0.86	0.92	0.13	0.17	0.15
K	0.92	0.73	0.81	0.47	0.16	0.24
L	0.5	0.86	0.63	0.33	0.83	0.48
M	0.75	0.64	0.69	0.8	0.46	0.59
N	0.58	0.7	0.64	0.33	0.62	0.43
O	1.0	0.92	0.96	0.8	0.75	0.77
P	0.75	0.9	0.82	0.6	0.75	0.67

Q	1.0	0.75	0.86	0.73	0.73	0.73
R	0.67	1.0	0.8	0.2	0.75	0.32
S	1.0	0.75	0.86	0.73	0.28	0.41
T	0.92	0.92	0.92	0.6	0.75	0.67
U	0.67	0.89	0.76	0.27	0.44	0.33
V	0.83	1.0	0.91	0.47	0.7	0.56
W	0.92	0.65	0.76	0.8	0.52	0.63
X	0.75	0.82	0.78	0.47	1.0	0.64
Y	0.92	0.73	0.81	0.33	0.33	0.33
Z	0.75	0.75	0.75	0.27	0.4	0.32

Pengujian pada preprocessing sharpening menghasilkan akurasi terbaik dibandingkan raw data maupun metode preprocessing lainnya. F1-Score menunjukkan nilai 0.94 untuk Seen Data dan 0.83 untuk Unseen Data. Hal ini menunjukkan bahwa sharpening dapat membantu model dalam memperjelas fitur yang akan diekstraksi dan digunakan dalam prediksi. Hasil pengujian sharpening dapat dilihat pada tabel 6.6.

**Tabel 6.6 Dataset Sendiri - Sharpening**

Class	Seem Test Data			Unseen Test Data		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Overall	<b>0.94</b>	<b>0.94</b>	<b>0.94</b>	<b>0.83</b>	<b>0.83</b>	<b>0.83</b>
Nothing	1.0	1.0	1.0	1.0	1.0	1.0
A	1.0	1.0	1.0	1.0	1.0	1.0
B	1.0	0.92	0.96	1.0	1.0	1.0
C	1.0	0.8	0.89	0.93	0.74	0.82

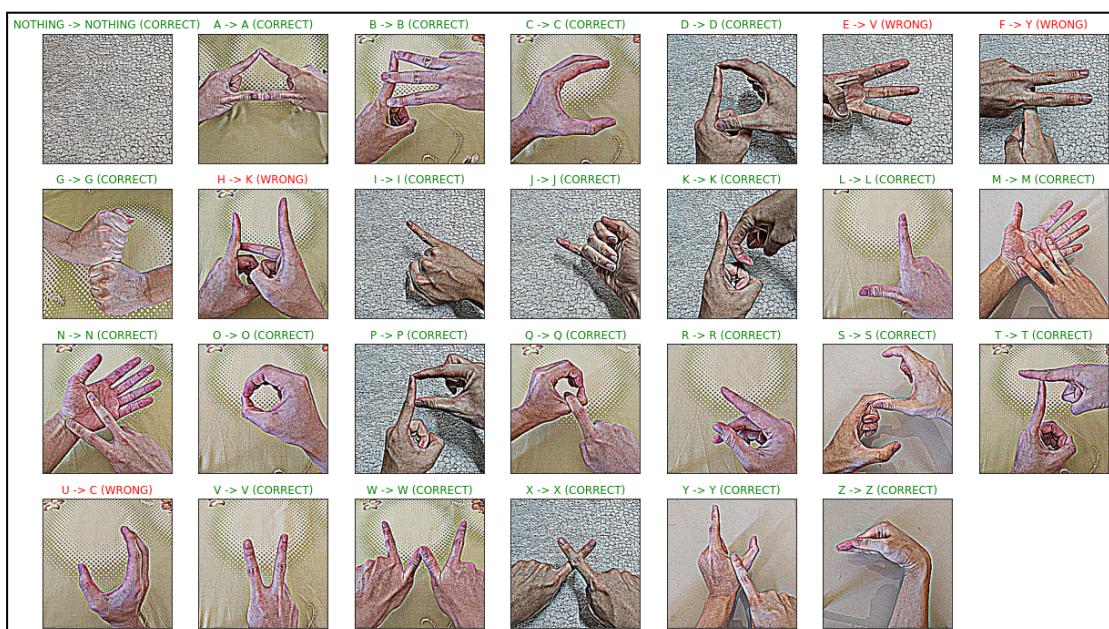
D	1.0	1.0	1.0	0.6	0.75	0.67
E	1.0	1.0	1.0	0.13	1.0	0.24
F	0.75	1.0	0.86	0.0	0.0	0.0
G	1.0	1.0	1.0	1.0	0.94	0.97
H	1.0	1.0	1.0	0.93	1.0	0.97
I	0.92	1.0	0.96	0.73	1.0	0.85
J	1.0	0.92	0.96	1.0	0.65	0.79
K	1.0	1.0	1.0	0.93	0.88	0.9
L	0.92	0.85	0.88	1.0	0.79	0.88
M	0.83	0.71	0.77	0.8	0.8	0.8
N	0.75	0.82	0.78	0.93	0.7	0.8
O	1.0	0.92	0.96	0.93	1.0	0.97
P	0.92	1.0	0.96	0.87	0.87	0.87
Q	1.0	1.0	1.0	0.6	1.0	0.75
R	0.92	1.0	0.96	0.87	1.0	0.93
S	0.83	0.77	0.8	0.67	0.67	0.67
T	1.0	1.0	1.0	1.0	1.0	1.0
U	0.83	1.0	0.91	0.67	0.77	0.71
V	0.83	1.0	0.91	1.0	0.52	0.68
W	0.83	1.0	0.91	0.87	1.0	0.93
X	1.0	0.92	0.96	1.0	1.0	1.0
Y	1.0	0.92	0.96	1.0	0.58	0.73
Z	0.92	0.85	0.88	1.0	0.94	0.97

Pada tabel 6.7 dipaparkan perbandingan performa antara metode preprocessing yang digunakan dalam pengujian. Preprocessing sharpening menghasilkan model dengan performa terbaik.

**Tabel 6.7 Dataset Sendiri - Preprocessing Comparison**

Preprocessing	Seen Test Data			Unseen Test Data		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
Raw	0.91	0.91	0.91	0.81	0.81	0.81
Grayscale	0.9	0.9	0.9	0.7	0.7	0.7
Edge Detection	0.27	0.27	0.27	0.12	0.12	0.12
Segmentation	0.84	0.84	0.84	0.49	0.49	0.49
<b>Sharpening</b>	<b>0.94</b>	<b>0.94</b>	<b>0.94</b>	<b>0.83</b>	<b>0.83</b>	<b>0.83</b>

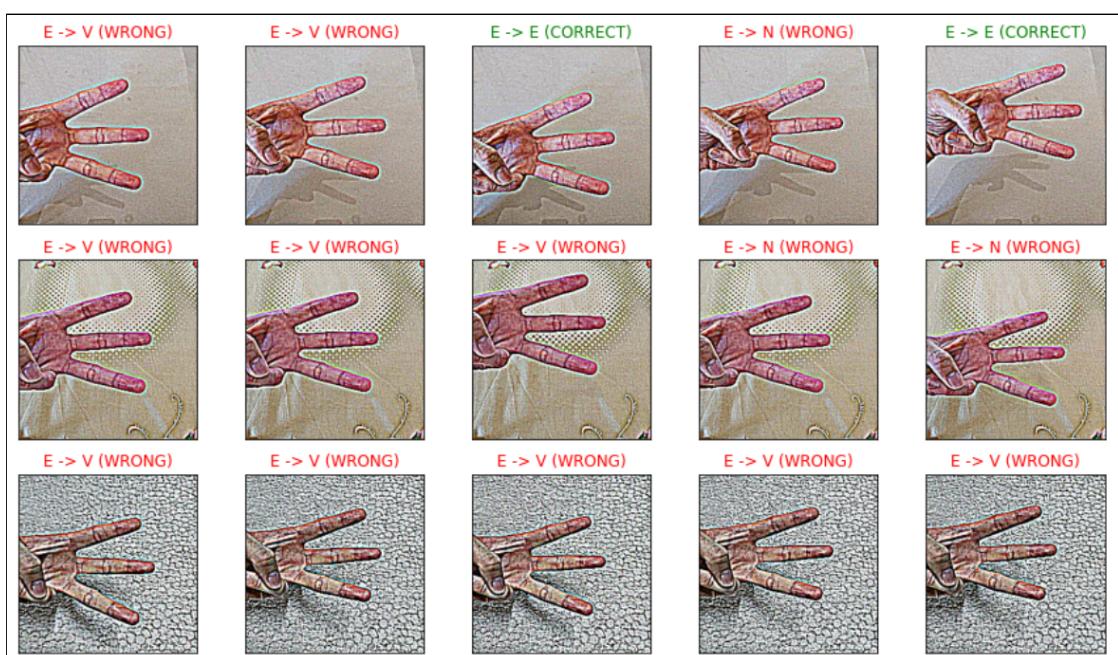
Pada gambar 6.1 diperlihatkan contoh beberapa prediksi dengan menggunakan image yang telah melalui preprocessing berupa image sharpening. Gambar yang ditunjukkan terdiri atas satu gambar per kelas yang dipilih secara random dari Unseen Data.



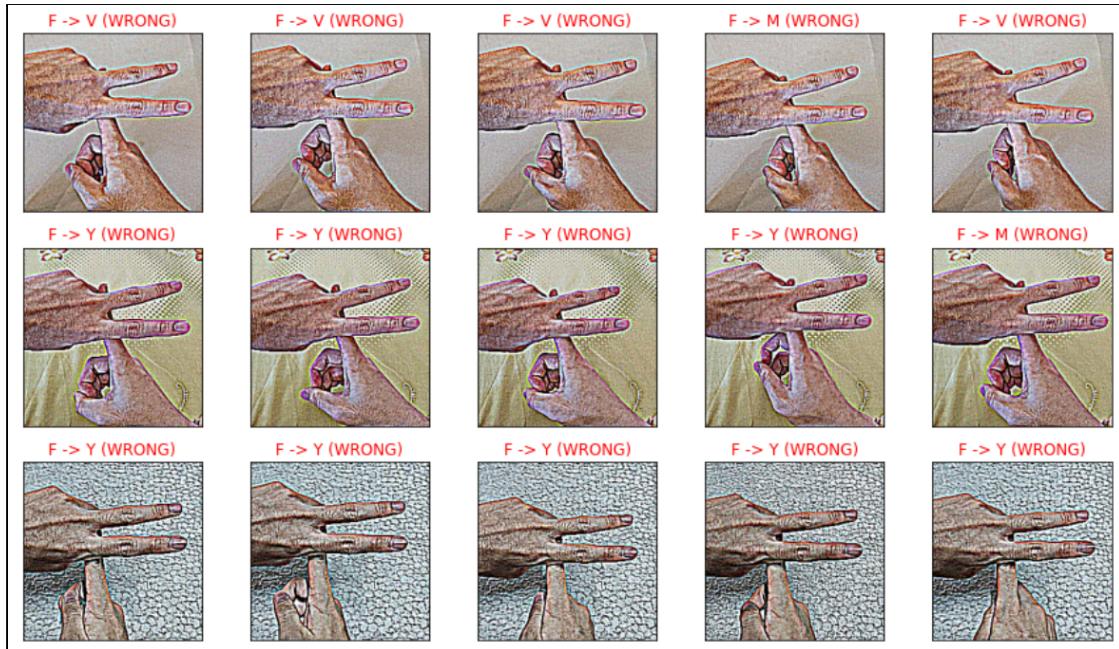
**Gambar 6.1 Contoh Prediksi Dataset Sendiri - Sharpening - Unseen Test Data**

## VII. Conclusions

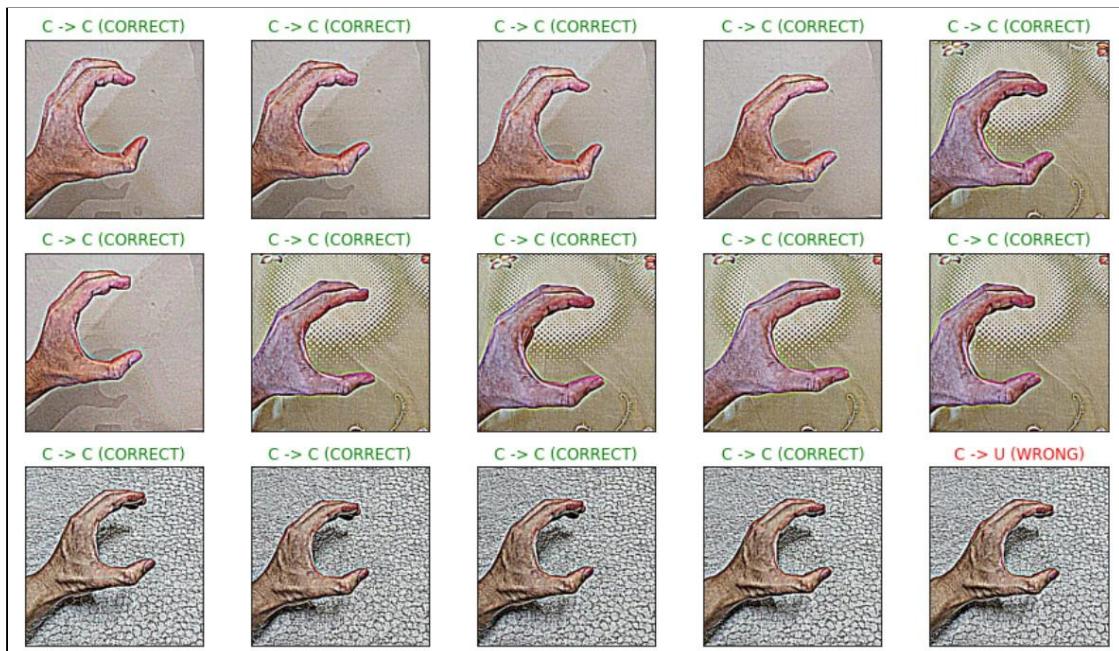
Penggunaan dataset yang bervariasi akan menghindarkan model dari *overfitting*. Penggunaan EfficientNet B0 pada dataset foto *Indonesian sign language* memiliki hasil yang lebih bagus dengan konfigurasi unfreeze 20 layers paling atas karena dataset yang digunakan tidak memiliki hubungan yang terlalu erat dengan dataset ImageNet. Dengan demikian, model dapat tetap memanfaatkan weights yang sudah ada dengan sedikit penyesuaian dari dataset baru. Berdasarkan hasil eksperimen perbandingan beberapa metode preprocessing yang dilakukan, didapati bahwa sharpening menghasilkan precision, recall, dan f1-score yang paling tinggi seperti tertulis pada tabel 6.7. Setiap huruf rata-rata dapat terdeteksi dengan baik kecuali huruf E dan huruf F (gambar 7.1 dan 7.2). Namun, terdapat juga kasus di mana model kesulitan dalam membedakan huruf yang memiliki struktur yang mirip, seperti antara huruf C dan U (gambar 7.3 dan 7.4) atau pun antara huruf M dan N (gambar 7.5 dan 7.6).



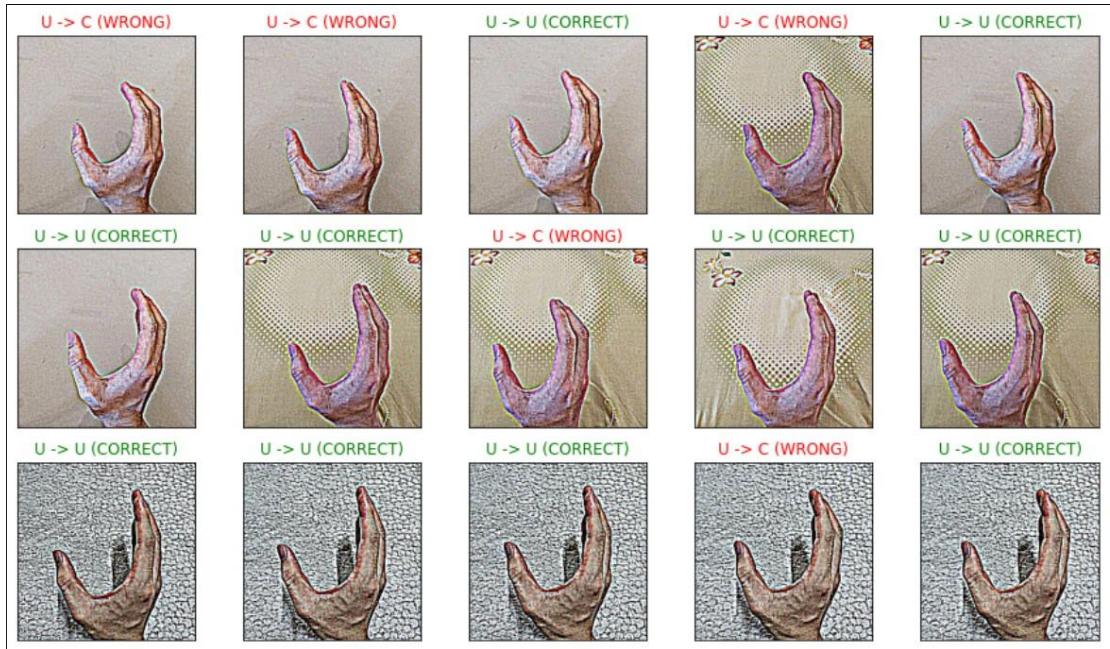
Gambar 7.1 Contoh Kesalahan Klasifikasi Huruf E



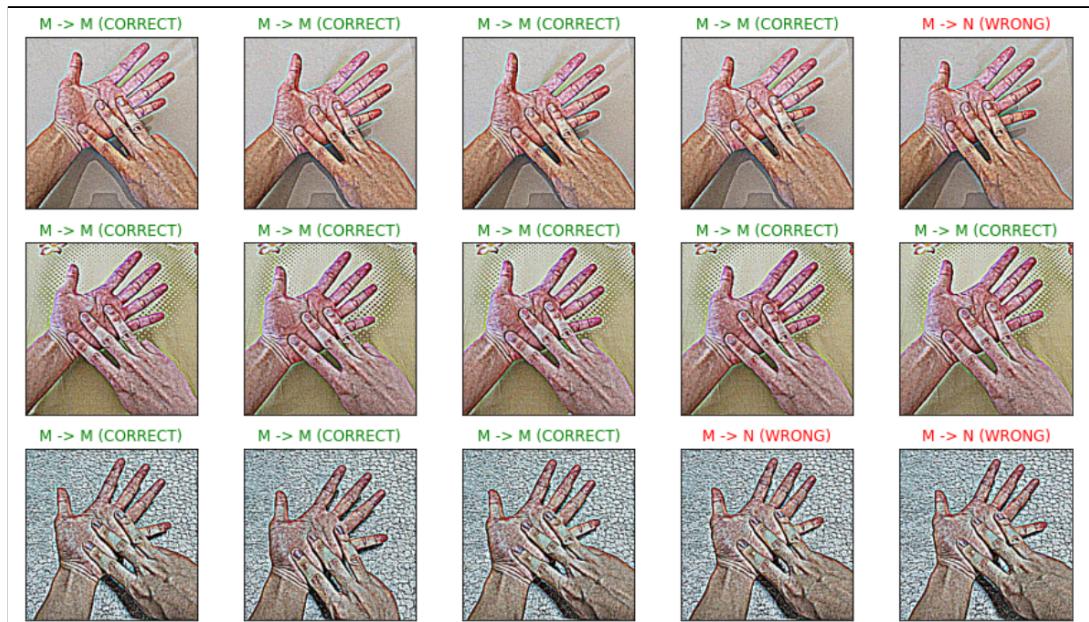
**Gambar 7.2 Contoh Kesalahan Klasifikasi Huruf F**



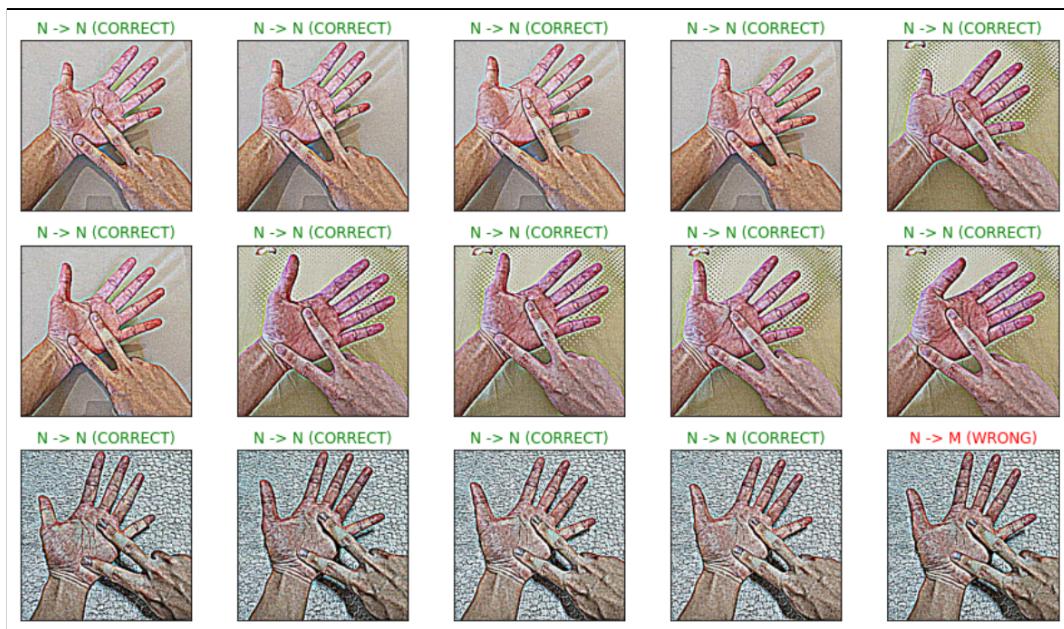
**Gambar 7.3 Contoh Kesalahan Klasifikasi Huruf C menjadi Huruf U**



**Gambar 7.4 Contoh Kesalahan Klasifikasi Huruf U menjadi Huruf C**



**Gambar 7.5 Contoh Kesalahan Klasifikasi Huruf M menjadi Huruf N**



**Gambar 7.6 Contoh Kesalahan Klasifikasi Huruf N menjadi Huruf M**

### VIII. Academic Reflection

Untuk memperoleh akurasi yang lebih baik, terdapat beberapa saran atau ide-ide yang dapat diimplementasikan di penelitian berikutnya. Berikut merupakan ide-ide tersebut:

- **Tomy Widjaja - C14180006**

Menggunakan pendekatan yang berbeda dalam bentuk input (misalnya menggunakan sensor Kinect untuk menangkap gesture / sequence of image). Dengan demikian, bentuk input menjadi bentuk data lain seperti gestur tangan dan jarak atau sudut antara jari-jari sehingga menghasilkan data yang tidak bergantung dengan warna dan background.

- **Enrico Hartanto - C14180013**

Penambahan jumlah dan variasi dataset yang lebih sesuai dengan kondisi real (Berbagai warna background, posisi tangan, pencahayaan, warna kulit, aksesoris tangan, dan sebagainya).

- **Henry Wicaksono - C14180014**

Preprocessing image input agar posisi tangan berada di tengah dan memenuhi keseluruhan image. Implementasi preprocessing ini dapat

dilengkapi dengan modul lokalisasi letak tangan agar dapat diimplementasikan pada kondisi real.

- **Andrew Firman - C14180029**

Preprocessing penambahan red dot pada ujung jari, khususnya untuk ibu jari karena pergerakan ibu jari yang terbatas sehingga lebih dapat mengenali orientasi secara mutlak.

- **Gregorius Nicholas - C14180046**

Menambah preprocessing pada image input dengan melakukan segmentasi gambar tangan dari background (Background removal / segmentation / image matting).

- **Gregorius Kevin - C14180057**

Menambahkan preprocessing median filter setelah preprocessing sharpening agar dapat menghilangkan salt and pepper noise.

## IX. What We Learned

Setiap anggota kelompok yang terlibat telah mendapatkan beberapa pelajaran dan pengalaman baru melalui proses pengerjaan proyek ini. Berikut adalah pemaparan beberapa pelajaran dan pengalaman yang didapat tersebut:

- **Tomy Widjaja - C14180006**

Melalui pengerjaan proyek akhir Deep Learning ini saya mendapatkan banyak pengalaman dan mempelajari berbagai hal mulai dari preprocess image, transfer learning, konfigurasi parameter, dan menerapkan evaluasi. Saya menyadari bahwa untuk kasus deteksi sign language sangat membutuhkan dataset yang bervariasi agar dapat digunakan secara nyata, karena jika hanya bergantung pada satu tangan orang untuk training maka model akan mengalami overfitting. Saya juga belajar bahwa untuk membuat model yang bagus harus mempertimbangkan berbagai kondisi untuk aplikasi secara real berbeda-beda (perbedaan cahaya, kamera, dll) sehingga sangat membutuhkan preprocessing. Dengan adanya transfer learning, saya dapat memanfaatkan weights yang sudah ada sehingga tinggal melakukan training weights dari beberapa layer bagian atas (atau fully connected layer nya) agar dapat menyesuaikan dengan dataset yang digunakan.

- **Enrico Hartanto - C14180013**

Pengerjaan proyek UAS Deep Learning ini adalah pertama kalinya saya belajar dan mencoba melakukan transfer learning, khususnya menggunakan EfficientNet. Saya menjadi tahu bahwa proses transfer learning berbeda dengan sekedar menggunakan pre-trained model. Pada transfer learning dapat ditentukan layer mana saja yang ingin ditrain ulang untuk menyesuaikan dengan dataset yang digunakan. Saya menyadari pentingnya variasi dari dataset apabila ingin mendapatkan model yang mampu mengklasifikasi input yang juga bervariasi dengan benar (tidak overfit). Saya juga mendapati bahwa preprocessing input sangat diperlukan agar data yang akan diklasifikasi lebih seragam dengan data yang digunakan untuk training, terutama apabila data input berasal dari kondisi real.

- **Henry Wicaksono - C14180014**

Melalui pengerjaan proyek akhir kelas Deep Learning ini, saya mendapatkan pengalaman untuk mencoba melakukan transfer learning pada EfficientNet sebagai salah satu model CNN. Saya juga mempelajari bahwa variasi data yang digunakan saat training sangatlah berpengaruh terhadap model. Data training yang kurang bervariasi justru akan membuat model menjadi overfit. Selain itu, saya juga mempelajari bagaimana pengaruh berbagai metode preprocessing terhadap kemampuan model dalam melakukan klasifikasi. Dengan mencoba menganalisa hasil performa model dan hubungannya dengan dataset yang digunakan, saya juga secara tidak langsung mempelajari karakteristik dan ciri khas dari setiap metode preprocessing yang digunakan tersebut.

- **Andrew Firman - C14180029**

Pada pengerjaan proyek UAS Indonesian Sign Language kali ini saya mempelajari bagaimana proses melakukan training *deep learning model* dengan *transfer learning* pada model EfficientNet dan proses pengumpulan dan pembuatan dataset yang harus memperhatikan dan sesuai terhadap kondisi nyata atau *real-world usage scenario*. Dalam proses training model saya juga mempelajari pengaruh dari *batch size* per *epoch* terhadap *waktu* per *epoch* pada training. Saya juga mempelajari pentingnya melakukan *preprocessing* yang tetap terhadap sebuah dataset. Pada proses *preprocessing*, bila proses dilakukan tidak tepat maka performa dari *model*

semakin menurun, tetapi bila *preprocessing* yang dilakukan tepat maka hasil *model* akan membaik.

- **Gregorius Nicholas - C14180046**

Setelah mengerjakan tugas akhir deep learning ini saya mempelajari mengenai proses transfer learning sebuah neural network. Dalam proses transfer learning tersebut saya baru memahami mengenai *fine tuning* dari neural network yang sudah ada sebelumnya seperti EfficientNet dan bagaimana pengaruh *fine tuning* pada hasil implementasi dalam proyek ini. Setelah melakukan fine tuning neural network maka hasil yang dihasilkan menunjukkan peningkatan yang cukup signifikan. Selain itu saya juga belajar mengenai pentingnya variasi pada dataset training sebuah neural network. Jika pada dataset training gambar kurang bervariasi maka hasilnya prediksi / klasifikasi -nya tidak akan optimal, biasanya disebut garbage in garbage out. Saya juga belajar mengenai berbagai macam *preprocessing* yang dapat digunakan pada image yang akan digunakan dalam training neural network.

- **Gregorius Kevin - C14180057**

Setelah mengerjakan tugas akhir dalam mata kuliah deep learning ini saya dapat mempelajari salah satu proses transfer learning yaitu EfficientNet yang sebelumnya hanya pernah saya baca pada saat mata kuliah Metpen. Selain itu mengenai topik yang kelompok kami ambil, saya mendapatkan bahwa data training yang bervariasi tinggi sangat penting karena dapat membuat proses training jadi lebih baik,gampang dan juga menghindari overfitting. Selain itu dalam pada saat mengerjakan topik yang berhubungan dengan gambar atau video, sangat penting bahwa kita mengerti banyak aspek yang dapat mempengaruhi proses pengolahan data, baik dari sudut kamera, noise pada gambar, nuansa background dan juga dataset yang bervariasi. Karena itu kita harus mengolah dataset yang didapatkan dengan baik, bukan hanya langsung memakai dataset tersebut saja.

## X. References

- Yunanda, A. B., et al. (2018). Pengenalan Bahasa Isyarat Indonesia (BISINDO) Untuk Karakter Huruf Dengan Menggunakan Microsoft Kinect. *Fountain of Informatics Journal*, 3(2), 41-45. <http://dx.doi.org/10.21111/fij.v3i2.2469>

Gupta, H., et al. (2019) . Sign Language Interpreter using Deep Learning. *Retrieved from* [\*https://github.com/harshbg/Sign-Language-Interpreter-using-Deep-Learning\*](https://github.com/harshbg/Sign-Language-Interpreter-using-Deep-Learning)