

Deteksi Aktivitas Pelajar menggunakan Pembelajaran Mesin Berbasis Jaringan Saraf Konvolusional guna Mengenal Karakter Pelajar

Raden Muhammad Kevin Ardiansyah^{1*}

S1 Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Negeri Malang,
raden.muhammad.2003126@students.um.ac.id¹

*Email : raden.muhammad.2003126@students.um.ac.id

Abstrak

Pendidikan adalah salah satu dari beberapa bidang dimana kecerdasan buatan akan menciptakan dampak yang sangat besar. Sangat penting untuk dipahami bahwa kecerdasan buatan bukanlah ancaman bagi para guru, kecerdasan buatan tidak dimaksudkan untuk menggantikan guru tetapi untuk memberikan pengalaman pendidikan yang lebih baik untuk para pelajar dan para guru. Salah satu variabel penting untuk melakukan pendekatan kepada para pelajar adalah aktivitasnya. Guna mendapatkan pendekatan yang lebih baik kepada pelajar, guru dapat mendeteksi aktivitas pelajar dengan menggunakan penerapan dari pembelajaran mesin yang merupakan bagian spesifik dari kecerdasan buatan. Harapannya adalah dengan menerapkan pembelajaran mesin, dapat diperoleh pendekatan yang lebih akurat dibandingkan metode lainnya. Pada penelitian ini digunakan model pembelajaran mesin berbasis jaringan saraf konvolusional pada himpunan data berupa gambar tindakan manusia yang dikumpulkan oleh Komunitas AI Planet. Hasil deteksi tindakan manusia pada data uji coba menunjukkan bahwa model VGGNet-16 (model jaringan saraf konvolusional sangat mendalam yang dikemukakan oleh Zisserman A. dan Simonyan K.) mampu memprediksi jenis tindakan manusia dengan akurat. Kesimpulan dari penelitian ini adalah hasil perolehan model pembelajaran mesin berbasis jaringan saraf konvolusional ini dapat diterapkan pada CCTV guna mendeteksi aktivitas pelajar yang nantinya bisa dikelola untuk kepentingan pengenalan karakter maupun keperluan lainnya.

Kata kunci: Pendidikan, deteksi, aktivitas pelajar, pembelajaran mesin, pengelompokan gambar

PENDAHULUAN

Pendidikan adalah salah satu dari beberapa bidang dimana kecerdasan buatan akan menciptakan dampak yang sangat besar. Sangat penting untuk dipahami bahwa kecerdasan buatan bukanlah ancaman bagi para guru, kecerdasan buatan tidak dimaksudkan untuk menggantikan guru tetapi untuk memberikan pengalaman pendidikan yang lebih baik untuk para pelajar dan para guru. Sampai batasan tertentu, kecerdasan buatan dapat membantu mengenali dan ikut memikirkan keseluruhan sistem pendidikan. Hwang et al. (2020) menyatakan bahwa dengan bantuan teknologi kecerdasan buatan, yang mensimulasikan kecerdasan manusia untuk membuat prediksi, sistem komputer dapat memberikan dukungan yang dipersonalisasi kepada pelajar maupun guru atau membuat kebijakan dalam membuat keputusan [1]. Hal tersebut dapat mempermudah guru untuk mengendalikan dan mengolah strategi dalam menyusun metode pembelajaran guna mendapatkan pendekatan yang lebih baik kepada pelajar. Salah satu pendekatan kepada pelajar yang cukup ampuh yaitu dengan mengenal kepribadian atau karakter mereka. Dengan mengenali karakter pelajar, guru akan dengan mudah menciptakan lingkungan yang dapat menumbuhkan potensi pelajar secara lebih maksimal. Sehingga, para pelajar akan tumbuh menjadi pribadi yang berperilaku baik, cerdas, serta sehat.

Melacak aktivitas pelajar di lingkungan sekolah merupakan tantangan yang cukup sulit bagi sebagian besar guru dan pegawai sekolah apabila dikerjakan setiap hari. Banyaknya aktivitas pelajar dalam satu ruang memerlukan banyak pegawai untuk melacak mereka sehingga bantuan dari sistem kecerdasan buatan sangat diperlukan. Salah satu bagian spesifik dari kecerdasan buatan yang sangat mumpuni untuk menangani permasalahan ini adalah pembelajaran mesin, yaitu ilmu yang memperlakukan komputer untuk belajar tanpa diprogram secara eksplisit. Harapannya adalah pemodelan berbasis pembelajaran mesin dapat mendeteksi beragam jenis aktivitas pelajar dengan akurat sehingga selanjutnya para guru dapat mengenal karakter masing-masing pelajar dan membuat strategi pembelajaran yang lebih ampuh. Model pembelajaran mesin berbasis jaringan saraf konvolusional seperti VGGNet-16 yang telah dilatih menggunakan data *ImageNet* (terdiri atas lebih dari 14 juta gambar yang terklasifikasi menjadi hampir 1000 label, termasuk layar monitor, hewan-

hewan, alat-alat tulis, *mouse*, dan lain-lain) dapat menjadi solusi untuk kasus pengelompokan aktivitas manusia karena arsitekturnya yang mendukung 16 lapis topologi jaringan khusus untuk kasus kompleks.

Penelitian ini ditujukan untuk merancang model pembelajaran mesin dengan pendekatan kasus klasifikasi pada aktivitas pelajar menggunakan model yang dilatih terhadap himpunan data aktivitas manusia. Kode dari penelitian ini dapat diakses (untuk lihat) di laman *GitHub* <https://github.com/RMKevinA/Belajar-HAR> pada file HAR-CNN.ipynb.

METODE

A. Rancangan Penelitian

Platform yang digunakan dalam penelitian ini adalah *Google Colaboratory* dengan dukungan GPU tipe T4 yang berjalan di *cloud*. Bahasa pemrograman yang digunakan adalah *Python* dan *library* dasar yang diperlukan adalah *NumPy* (mendukung himpunan dan matriks multidimensi yang besar serta mengoperasikan fungsi matematika tingkat tinggi), *Pandas* (untuk manipulasi dan analisis data), dan *Matplotlib and Plotly* (untuk membuat visualisasi yang statis, animatif, dan interaktif dalam *Python*). Dalam pengembangan model pembelajaran mesin dan model pembelajaran mendalam diperlukan *library* khusus dan berikut adalah beberapa *library* yang dapat digunakan dalam penelitian ini.

i. *Scikit-learn*

Scikit-learn adalah *library* pembelajaran mesin yang sangat populer yang dibangun menggunakan *NumPy* dan *SciPy*. *Scikit-learn* mendukung sebagian besar algoritma pembelajaran *supervised* dan *unsupervised* klasik, dan juga dapat digunakan untuk *data mining*, pemodelan, dan analisis. Desain sederhana *Scikit-learn* menawarkan *library* yang mudah digunakan oleh para pemula dalam lingkup pembelajaran mesin.

ii. *TensorFlow*

Open-source Python library milik *TensorFlow* spesialis dalam *differentiable programming*, artinya *library* ini dapat secara otomatis menghitung turunan fungsi dalam *high-level language*. Model pembelajaran mesin dan pembelajaran mendalam mudah dikembangkan dan dievaluasi dengan arsitektur dan kerangka kerja *TensorFlow* yang fleksibel.

iii. *PyTorch*

PyTorch adalah *python library* pembelajaran mesin *open-source* berdasarkan kerangka kerja bahasa pemrograman *C*, yakni *Torch*. *PyTorch* secara umum digunakan dalam penerapan pembelajaran mesin yang melibatkan *natural language processing* (NLP) maupun *computer vision*. *PyTorch* dikenal sangat cepat dalam mengeksekusi kumpulan data dan grafik dalam jumlah banyak dan padat.

iv. *Keras*

Keras adalah *Python library* yang dirancang khusus untuk mengembangkan jaringan saraf untuk model pembelajaran mesin. *Keras* dapat berjalan di atas *Theano* dan *TensorFlow* untuk melatih jaringan saraf. *Keras* dikenal fleksibel, ringan, dan mudah digunakan, serta mudah diintegrasikan dengan berbagai fungsi.

Pada penelitian ini akan digunakan *library Keras* yang berjalan diatas *TensorFlow* karena *library* tersebut sangat memudahkan proses mengembangkan model pembelajaran mesin berbasis jaringan saraf mendalam (bahkan sangat mendalam) yang cocok untuk mengolah data berupa gambar. Penelitian dimulai dengan mengunduh himpunan data aktivitas manusia yang telah dikumpulkan (tersedia pada laman web *AI Planet*) dan mengimpor seluruh *library* yang dibutuhkan.

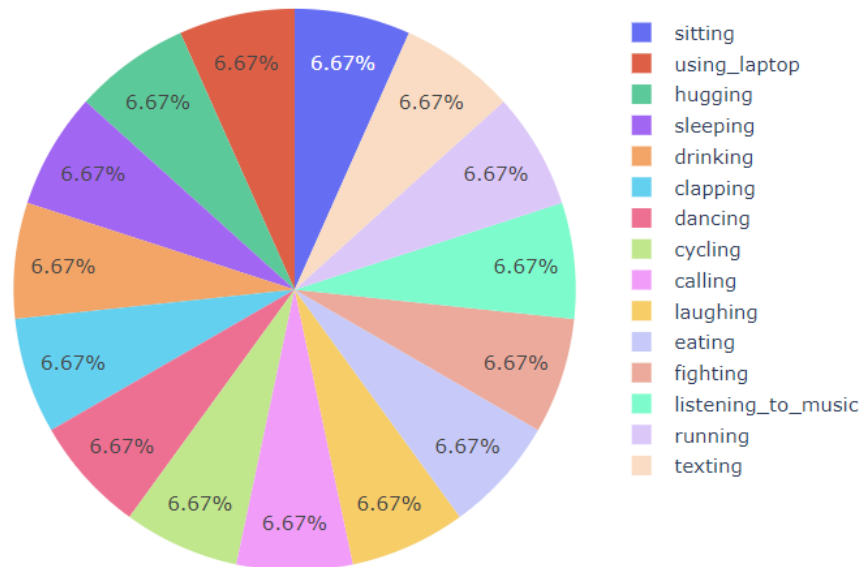
B. Data Pemodelan

Tahap selanjutnya adalah mengeksplorasi data dari hasil pengumpulan yang dilakukan oleh komunitas *AI Planet* yang nantinya akan digunakan untuk merancang dan mengevaluasi model. Dari hasil pengumpulan terdapat dua jenis data yaitu data *training* dengan 12600 gambar dan data *test* dengan 5400 gambar. Gambar yang terdapat pada data *training* dan data *test* merupakan gambar yang berbeda-beda karena untuk pengujian

model diperlukan data yang asing (tidak dikenali oleh model yang telah dilatih sebelumnya) sehingga dapat diketahui tingkat akurasi dalam memprediksi gambar yang baru. Berkas juga dilengkapi dengan *csv file* untuk masing-masing data *training* dan *test* sebagai keterangannya (dua kolom yakni *filename* dan *label* untuk data *training* dan satu kolom yakni *filename* untuk data *test*). Gambar pada data *training* dikelompokkan menjadi 15 label yakni *sitting*, *using_laptop*, *hugging*, *sleeping*, *drinking*, *clapping*, *dancing*, *cycling*, *calling*, *laughing*, *eating*, *fighting*, *listening_to_music*, *running*, dan *texting*. Pada pemodelan hanya akan menggunakan data *training* dan untuk menguji tingkat akurasi terhadap gambar yang asing akan digunakan data *test*.

C. Analisis Data

Berikut adalah visualisasi distribusi data *training*.



Gambar 1. Diagram Distribusi dari Jumlah Data berdasarkan Lima Belas Label di Data *Training*

Dari visualisasi distribusi data di atas dapat dilihat bahwa himpunan data seimbang atau *balanced dataset*. Menurut studi yang dilakukan Pecorelli et al. (2019), suatu model akan menjadi lebih akurat ketika teknik penyeimbangan terbaik digunakan [2]. Secara tersirat, model dapat memiliki tingkat akurasi yang tinggi apabila himpunan data yang digunakan untuk melatih model tersebut memiliki distribusi data yang seimbang. Sehingga, pada tahap selanjutnya akan dilakukan pengolahan data sebelum digunakan sebagai masukan model.

D. Pengolahan Data

Data berupa gambar yang diperoleh sebelumnya memiliki ukuran piksel yang berbeda-beda sehingga diperlukan untuk mengubah ukuran piksel masing-masing gambar tersebut. Gambar-gambar tersebut akan diubah ukurannya menjadi 160×160 RGB (Red Green Blue) sehingga gambar dapat dikenal dengan cukup jelas dan dengan ukuran yang tidak terlalu besar karena ukuran gambar yang besar akan menambah waktu pelatihan dari model. Selanjutnya diterapkan teknik *one-hot encoding* pada label dari setiap gambar menggunakan metode *to_categorical* yang disediakan oleh *Keras* untuk mengubah variabel kategorikal menjadi bentuk numerik yang terdiri dari angka '0' dan '1' yang dapat diolah oleh algoritma pembelajaran mesin. Berikut merupakan hasil sebelum dan sesudah proses *one-hot encoding* untuk label 10 data pertama.

0	sitting	array([[1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
1	using_laptop	[0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
2	hugging	[0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
3	sleeping	[0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
4	using_laptop	[0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
5	sleeping	[0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
6	drinking	[0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
7	hugging	[0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
8	clapping	[0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
9	dancing	[0., 0., 0., 0., 0., 0., 1., 0., 0., 0., 0., 0., 0., 0., 0., 0.]
Name: label, dtype: object		dtype=float32)

Gambar 2. Hasil Proses *One-hot Encoding* pada Label 10 Data Pertama

E. Pemodelan

Pada tahap pembuatan model untuk mendeteksi aktivitas pelajar ini, akan digunakan model jaringan saraf mendalam atau *deep neural networks* (DNN) yang dalam kasus pengenalan gambar yaitu disebut sebagai jaringan saraf konvolusional atau *convolutional neural networks* (CNN). Alih-alih membuat CNN dari awal, terdapat cara legal yang memiliki keuntungan jauh lebih besar dan terbilang cukup praktis, yakni metode *transfer learning*. *Transfer learning* yang merupakan salah satu metode pada pembelajaran mesin atau *machine learning* di mana model yang sudah dilatih pada suatu tugas sebelumnya atau biasa disebut dengan *pre-trained model*, dapat digunakan untuk menyelesaikan tugas lain. Hal ini dapat dilakukan dengan cara mengambil beberapa lapisan topologi jaringan atau *layer* (kata *layer* akan digunakan untuk penamaan selanjutnya) konvolusi dari model yang sudah ada sebelumnya, dan membuat mereka supaya tidak ikut dilatih pada tahap pelatihan model, yaitu dengan cara mengunci *learned convolution layers* ke dalam model, lalu menambahkan beberapa DNN dibawah *layers* tersebut, di mana akan dilakukan *retrain* dengan data aktivitas manusia yang telah diolah pada tahap sebelumnya. Model DNN tambahan akan meliputi *flatten layers* dan *dense layers* dengan fungsi aktivasi. Terdapat beberapa jenis fungsi aktivasi cukup umum digunakan yakni sebagai berikut.

Diberikan $z = \vec{w} \cdot \vec{x} + b$, dimana \vec{w} merupakan vektor *weight*, \vec{x} merupakan vektor *feature*, dan b adalah bias,

- i. *Linear activation function*

$$\alpha(z) = z$$

- ii. *ReLU (Rectified Linear Unit) activation function*

$$\alpha(z) = \max(0, z)$$

- iii. *Tanh activation function*

$$\alpha(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

- iv. *Sigmoid/Logistic activation function* (2 kemungkinan nilai output)

$$\alpha(z) = \frac{1}{1 + e^{-z}}$$

- v. *Softmax activation function* (N kemungkinan nilai output)

$$\alpha_j(z) = \frac{e^{z_j}}{\sum_{k=1}^N e^{z_k}} ; j = 1, 2, 3, \dots, N.$$

Akan ditambahkan *dense layer* pada DNN dengan fungsi aktivasi *ReLU* untuk *dense hidden layer*nya dan *Softmax* untuk *output layer*nya (output model menjadi 15 label atau kelas sesuai data penelitian ini).

Kembali membahas mengenai *transfer learning*, telah dipaparkan pada Bagian Pendahuluan, *pre-trained model* dari VGG (*Visual Geometry Group, Department of Engineering Science, University of Oxford*) yang telah dilatih menggunakan data *ImageNet* dapat menjadi solusi untuk kasus pengelompokan aktivitas manusia. VGG menyajikan beberapa versi konfigurasi jaringan konvolusi sebagai berikut.

Tabel 1. Konfigurasi Jaringan Konvolusi (ConvNet) yang Disajikan oleh VGG

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224×224 RGB image)					
conv3-64	conv3-64 LRN	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 conv1-256	conv3-256 conv3-256 conv3-256	conv3-256 conv3-256 conv3-256 conv3-256
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 conv1-512	conv3-512 conv3-512 conv3-512	conv3-512 conv3-512 conv3-512 conv3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

Sumber: (Simonyan dan Zisserman, 2015: 3)

Masing-masing versi model di atas memiliki keunggulan yang berbeda-beda, namun untuk versi D (memiliki 16 *weight layers* dan arsitektur dengan *convolutional filters* 3×3 piksel) yang juga disebut dengan VGGNet-16 mendapatkan urutan 5 terbesar uji akurasi pada *ImageNet* yaitu 92,7%, sehingga sangat cocok apabila dilakukan metode *transfer learning* untuk tugas pengenalan aktivitas pelajar. Menurut Gambar 3., model tersebut memiliki masukan gambar RGB berukuran 224×224 sehingga perlu mengatur *hyperparameter input_shape*=(160, 160, 3) untuk DNN tambahan yang akan dibuat, mengingat data yang telah diolah untuk penelitian ini berupa gambar RGB berukuran 160×160 .

Sebetulnya terdapat banyak model populer untuk *large-scale image recognition* selain VGG, beberapa model paling populer diantaranya disajikan pada gambar tabel berikut.

Tabel 2. Perbandingan dengan Beberapa Metode Lainnya pada Lokalisasi ILSVRC (*ImageNet Large-Scale Visual Recognition Challenge*)

Method	top-5 val. error (%)	top-5 test error (%)
VGG	26.9	25.3
GoogLeNet (Szegedy et al., 2014)	-	26.7
OverFeat (Sermanet et al., 2014)	30.0	29.9
Krizhevsky et al. (Krizhevsky et al., 2012)	-	34.2

Sumber: (Simonyan dan Zisserman, 2015: 12)

Dapat diketahui bahwa metode yang dirancang VGG memiliki galat terkecil terhadap data validasi dan data *test* sehingga memilih model VGG sebagai target dari metode *transfer learning* pada kasus pengenalan aktivitas pelajar merupakan keputusan yang cukup tepat. Diketahui dari paparan detail proses implementasi model VGG, Simonyan dan Zisserman (2015) menyatakan bahwa model VGG mereka telah dilatih menggunakan GPU Nvidia Titan Black selama 2-3 pekan untuk setiap versi arsitekturnya [3]. Pelatihan model tersebut membutuhkan waktu yang lebih lama apabila tidak difasilitasi GPU berspesifikasi tinggi seperti yang digunakan oleh mereka. Tetapi dengan adanya metode *transfer learning*, tidak diperlukan pelatihan ulang

terhadap model sehingga dapat langsung diakses untuk diambil beberapa fiturnya dan ditambahkan pada model DNN yang dimiliki.

F. Pengujian Model

Pada tahap ini, telah ditentukan bahwa akan digunakan pengujian secara kustom. Pengujian ini dipilih karena gambar-gambar pada data *test* tidak memiliki label sehingga tidak dapat digunakan untuk memvalidasi tingkat akurasi model menggunakan algoritma. Pengujian kustom adalah menguji data secara acak lalu melihat hasil prediksi dan *probability* dari prediksi tersebut.

HASIL DAN PEMBAHASAN

Dari analisis data yang dilakukan, diperoleh informasi bahwa himpunan data seimbang atau *balanced dataset* sehingga model dapat memiliki tingkat akurasi yang tinggi. Setelah itu, data gambar diubah ukurannya menjadi 160×160 piksel RGB (Red Green Blue) dan data label gambarnya diproses *encode* menjadi *one-hot*.

A. Hyperparameter Tuning

Pada eksperimen yang telah dilakukan, diperoleh *hyperparameter* terbaik untuk model deteksi aktivitas pelajar sebagai berikut,

- i. *Pre-trained VGGNet-16 layers* (*layer.trainable=False* untuk semua *layer* di *Pre-trained VGGNet-16 layers*)
 - *include_top = False*,
 - *input_shape = (160,160,3)*,
 - *pooling = 'avg'*,
 - *classes = 15*,
 - *weights = 'imagenet'*
- ii. *Dense layer*
 - *units = 512*
 - *activation = 'relu'*
 - *use_bias = True*
- iii. *Dense_1 layer*
 - *units = 15*
 - *activation = 'softmax'*
 - *use_bias = True*
- iv. *model.compile*
 - *optimizer = tf.keras.optimizers.Adam()*
 - *loss = tf.keras.losses.CategoricalCrossentropy()*
 - *metrics = ['accuracy']*

B. Pelatihan Model

Setelah selesai membuat arsitektur untuk modelnya, dapat dilihat ringkasannya menggunakan metode *model.summary()* dan diperoleh sebagai berikut.

```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 512)	14714688
flatten (Flatten)	(None, 512)	0
dense (Dense)	(None, 512)	262656
dense_1 (Dense)	(None, 15)	7695

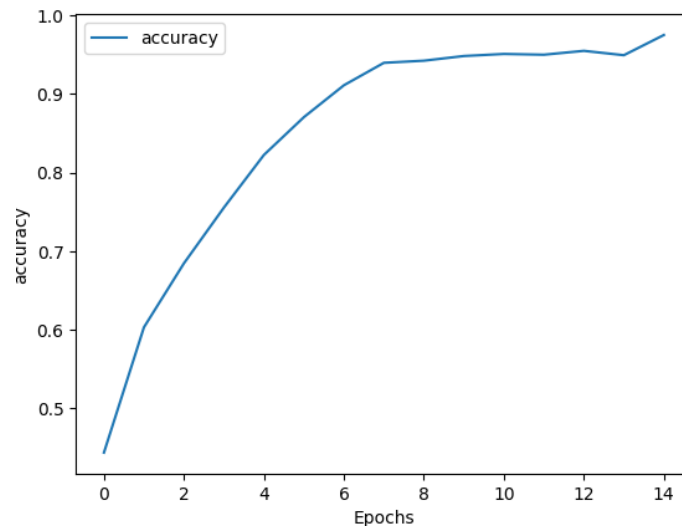
```

=====
Total params: 14,985,039
Trainable params: 270,351
Non-trainable params: 14,714,688
=====

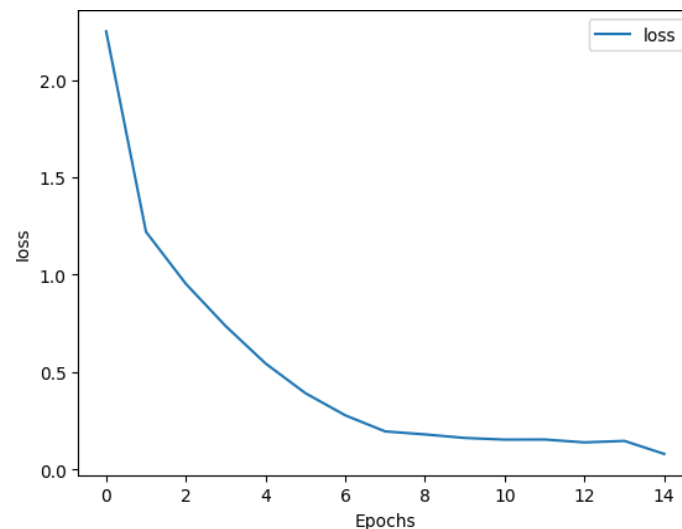
```

Gambar 3. Ringkasan Arsitektur Model

Selanjutnya dilakukan eksperimen pelatihan model dan diperoleh bahwa dengan menggunakan 15 *epochs*, model memiliki tingkat akurasi yang tinggi dan *loss* yang cukup rendah. *Epoch* dilakukan dalam jumlah secukupnya (sampai akurasi dan *loss* cukup bagus) untuk menghindari *overfitting* pada data *train*. Berikut merupakan grafik dari pelatihan model dengan 15 *epochs*.



Gambar 4. Grafik dari *Accuracy Metric* pada Model dengan 15 *Epochs*



Gambar 5. Grafik dari *Loss Metric* pada Model dengan 15 *Epochs*


```

Epoch 1/15
394/394 [=====] - 39s 69ms/step - loss: 2.2499 - accuracy: 0.4438
Epoch 2/15
394/394 [=====] - 26s 65ms/step - loss: 1.2206 - accuracy: 0.6033
Epoch 3/15
394/394 [=====] - 26s 66ms/step - loss: 0.9536 - accuracy: 0.6847
Epoch 4/15
394/394 [=====] - 27s 68ms/step - loss: 0.7365 - accuracy: 0.7556
Epoch 5/15
394/394 [=====] - 27s 68ms/step - loss: 0.5447 - accuracy: 0.8226
Epoch 6/15
394/394 [=====] - 27s 67ms/step - loss: 0.3921 - accuracy: 0.8707
Epoch 7/15
394/394 [=====] - 26s 67ms/step - loss: 0.2785 - accuracy: 0.9113
Epoch 8/15
394/394 [=====] - 27s 68ms/step - loss: 0.1959 - accuracy: 0.9398
Epoch 9/15
394/394 [=====] - 27s 68ms/step - loss: 0.1805 - accuracy: 0.9425
Epoch 10/15
394/394 [=====] - 27s 67ms/step - loss: 0.1623 - accuracy: 0.9485
Epoch 11/15
394/394 [=====] - 27s 68ms/step - loss: 0.1538 - accuracy: 0.9511
Epoch 12/15
394/394 [=====] - 27s 68ms/step - loss: 0.1543 - accuracy: 0.9502
Epoch 13/15
394/394 [=====] - 27s 68ms/step - loss: 0.1396 - accuracy: 0.9551
Epoch 14/15
394/394 [=====] - 27s 67ms/step - loss: 0.1471 - accuracy: 0.9495
Epoch 15/15
394/394 [=====] - 27s 68ms/step - loss: 0.0804 - accuracy: 0.9752

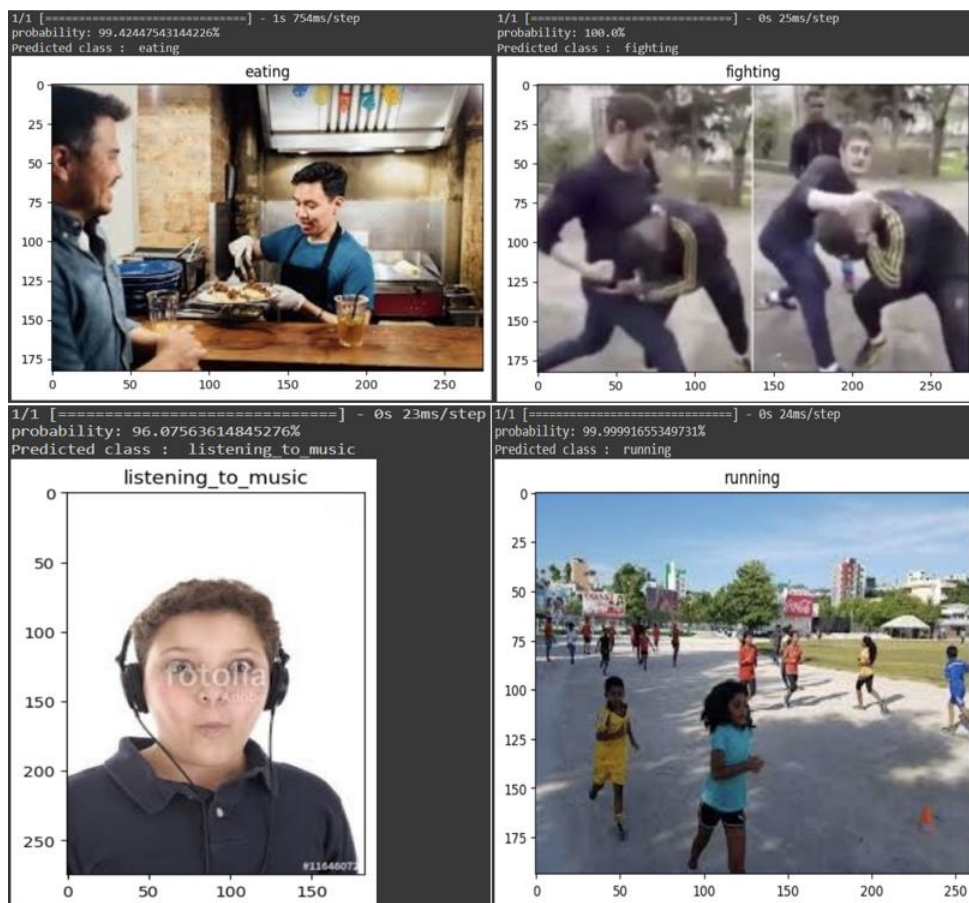
```

Gambar 6. *History* Pelatihan Model

Dapat dilihat di atas bahwa model memiliki tingkat akurasi sekitar 97% dengan *loss* sekitar 0.08 yang mana model telah terlatih dengan cukup baik pada data aktivitas manusia.

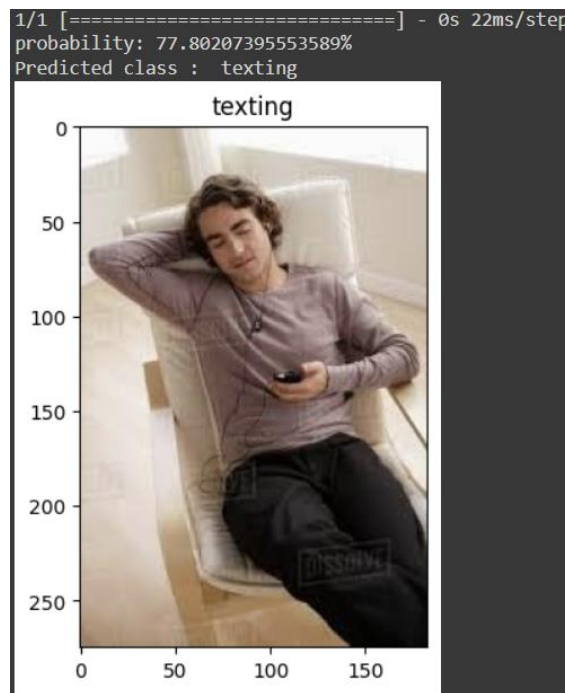
C. Pengujian Kustom

Setelah dilakukan pengujian kustom terhadap beberapa gambar pada himpunan data *test*, diperoleh hasil sebagai berikut.



Gambar 7. Hasil Prediksi pada Beberapa Gambar yang Belum Dilihat oleh Model

Namun terdapat beberapa gambar yang mungkin manusia juga bisa jadi melakukan kesalahan dalam memprediksi aktivitas manusia seperti berikut.



Gambar 8. Hasil Prediksi pada Gambar Seseorang yang sedang Tidur sekaligus Memegang Ponsel dan Mendengarkan Musik

Gambar 8. menampilkan aktivitas seseorang yang melakukan 3 aktivitas sekaligus sesuai label yang ditetapkan sebelumnya, yaitu *sleeping*, *listening_to_music*, dan *texting*. Manusia mungkin menyatakan bahwa aktivitas orang tersebut lebih condong ke *listening_to_music*, namun model memberikan output prediksi yang lebih condong ke *texting* dengan *probability*: 77,8%, hal ini cukup maklum terjadi.

Terdapat beberapa kasus yang menampilkan bahwa model bersifat *overfitting* pada suatu data tertentu, salah satunya adalah sebagai berikut.



Gambar 9. Hasil Prediksi pada Gambar Beberapa Orang yang sedang Berlari dan Beberapa Orang Lainnya sedang Bersepeda

Manusia mungkin menganggap aktivitas pada gambar tersebut adalah 50% sedang berlari dan 50% sedang bersepeda. Tetapi model hasil penelitian menganggap aktivitas tersebut sepenuhnya bersepeda (*probability: 100% on cycling*). Artinya model bersifat overfitting pada gambar bersepeda, hal ini kemungkinan terjadi karena gambar berlabel *cycling* pada data *train* memiliki variasi yang lebih banyak. Kemungkinan lain adalah terdapat gambar orang berlari yang *misllabeled* sebagai orang bersepeda.

PENUTUP

A. Kesimpulan

Berdasarkan hasil penelitian ini, diperoleh kesimpulan sebagai berikut,

- i. Hasil deteksi tindakan manusia pada data *test* atau uji coba menunjukkan bahwa model VGGNet-16 mampu memprediksi 15 jenis tindakan manusia dengan cukup akurat, meskipun terdapat sedikit kasus *overfitting* namun hal tersebut dapat dimaklumi mengingat manusia memungkinkan untuk salah dalam mengenali gambar tertentu
- ii. Hasil perolehan model pembelajaran mesin berbasis jaringan saraf konvolusional ini nantinya dapat diterapkan pada CCTV lewat komputer pengawas sekolah guna mendeteksi aktivitas pelajar yang nantinya bisa dikelola untuk kepentingan pengenalan karakter maupun keperluan lainnya.

B. Keterbatasan Penelitian

Penelitian ini hanya sebatas pembuatan model pembelajaran mesin berbasis jaringan saraf mendalam yang cukup akurat untuk mendeteksi aktivitas manusia dan dapat *dideploy* pada suatu sistem komputer. Harapannya adalah sistem kecerdasan buatan dengan model pembelajaran mesin ini dapat diterapkan di sekolah untuk mengenal apa saja aktivitas yang dilakukan pelajar. Kajian mengenai hubungan antara aktivitas fisik dengan karakter pelajar dan cara membuat aplikasi atau sistem pada CCTV sekolah tidak dibahas pada penelitian ini.

C. Saran

Saran untuk penelitian selanjutnya adalah sebagai berikut,

- i. Dapat diusulkan model-model pembelajaran mesin lainnya sebagai perbandingan dengan model pada penelitian ini,
- ii. Dapat dilakukan penelitian lebih lanjut tentang cara mengimplementasikan model pembelajaran mesin berbasis jaringan saraf mendalam ini ke dalam sistem kecerdasan buatan yang dapat dipasang pada CCTV lewat komputer pengawas sekolah.

DAFTAR RUJUKAN

- [1] G.-J. Hwang, H. Xie, B. W. Wah, dan D. Gašević, "Vision, challenges, roles and research issues of Artificial Intelligence in Education," *Computers and Education: Artificial Intelligence*, vol. 1, hlm. 100001, 2020, doi: 10.1016/j.caeai.2020.100001.
- [2] F. Pecorelli, F. Palomba, D. Di Nucci, dan A. De Lucia, "Comparing Heuristic and Machine Learning Approaches for Metric-Based Code Smell Detection," dalam *2019 IEEE/ACM 27th International Conference on Program Comprehension (ICPC)*, Montreal, QC, Canada: IEEE, Mei 2019, hlm. 93–104. doi: 10.1109/ICPC.2019.00023.
- [3] K. Simonyan dan A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition." arXiv, 10 April 2015. Diakses: 21 Juni 2023. [Daring]. Tersedia pada: <http://arxiv.org/abs/1409.1556>