

Deteksi Potensi Menyontek Menggunakan Feedforward Neural Network Pada Ujian Daring

Lymanto Hadibrata¹, Theresia Herlina Rochadiani²

^{1,2}Informatics, Faculty of Science and Technology, Pradita University
Jl. Gading Serpong Boulevard No.1 Tower 1, Kec. Klp. Dua, Kabupaten Tangerang, Indonesia

e-mail: lymanto.hadibrata@student.pradita.ac.id¹, theresia.herlina@pradita.ac.id²

Received : March, 2024

Accepted : May, 2024

Published : August, 2024

Abstract

Education in Indonesia is a pivotal factor that can contribute to the nation's progress towards becoming a developed country. However, the prevalence of cheating among students undermines the quality of education. This research aims to develop a deep learning model using the feedforward neural network for detecting potential cheating. The study utilizes a dataset of 51 videos that get from people, which are converted into coordinate points using MediaPipe Face Landmark and stored in CSV files. The dataset includes 7 classes: neutral, facing_up, facing_down, facing_left, facing_right, left_retina, and right_retina. The neutral class is the primary indicator for detecting non-cheating potential, while the left_retina and right_retina classes are also considered due to their relevance in reading questions. The facing_left, facing_right, facing_up, and facing_down classes are key indicators for detecting cheating potential. The research yields a model capable of predicting cheating potential with an accuracy of 91.6% using the feedforward neural network method. This model can be implemented in an online examination system.

Keywords: Detecting Potential Cheating, Education, Feedforward Neural Network, MediaPipe Face Landmark

Abstrak

Pendidikan di Indonesia merupakan salah satu faktor pendukung yang dapat menjadikan Indonesia menjadi negara maju. Akan tetapi masih banyak pelajar yang melakukan praktik menyontek sehingga menurunkan kualitas pendidikan di Indonesia. Untuk mengurangi praktik menyontek di Indonesia, penelitian ini bertujuan untuk membuat sebuah model deep learning dengan menggunakan metode feedforward neural network untuk mendeteksi potensi menyontek. Penelitian ini menggunakan 51 video dataset diperoleh dari orang-orang yang pada akhirnya diubah menjadi titik-titik koordinat menggunakan MediaPipe Face Landmark yang disimpan pada file CSV. Pada penelitian ini terdapat 7 class pada dataset yang sudah dibuat yaitu netral, hadap_atas, hadap_bawah, hadap_kiri, hadap_kanan, retina_kiri dan retina_kanan. Indikator utama yang paling menentukan untuk mendeteksi potensi tidak menyontek adalah class netral. Akan tetapi, class retina_kiri dan retina_kanan juga ikut berpartisipasi karena ada pertimbangan dari segi pembacaan soal. Indikator yang menentukan untuk mendeteksi potensi menyontek adalah class hadap_kiri, hadap_kanan, hadap_atas, dan hadap_bawah. Penelitian ini menghasilkan model yang dapat memprediksi potensi menyontek dengan akurasi sebesar 91.6% dengan menggunakan metode feedforward neural network. Dari model yang dihasilkan, dapat diimplementasikan kedalam sebuah sistem ujian daring.

Kata Kunci: Deteksi Potensi Menyontek, Feedforward Neural Network, MediaPipe Face Landmark, Pendidikan.

1. PENDAHULUAN

Pendidikan merupakan suatu pengalaman belajar yang terstruktur dalam bentuk pendidikan formal dan nonformal di dalam dan luar sekolah, yang berlangsung selamanya [1]. Pendidikan di Indonesia merupakan salah satu faktor pendukung yang dapat menjadikan Indonesia menjadi negara maju. Akan tetapi masih banyak kecurangan yang terjadi pada saat ujian, dari hasil penelitian terdahulu terdapat 69% dari 45 siswa yang melakukan aktifitas menyontek di SD Negeri Talkondo [2].

Menyontek dalam ujian membawa konsekuensi negatif yang signifikan. Hal ini memungkinkan siswa untuk memperoleh nilai tinggi dan keuntungan secara tidak adil, yang merugikan mereka yang tidak menyontek. Menyontek juga menghambat upaya untuk mengembangkan rasa tanggung jawab etis pada siswa dan dapat merusak reputasi lembaga pendidikan. Selain itu, menyontek berdampak buruk pada pembelajaran karena siswa melewatkan kesempatan belajar yang penting. Misalnya, siswa yang menyontek pada tes latihan online cenderung memiliki kinerja buruk pada ujian akhir. Selain itu, menyontek mencegah guru memberikan umpan balik yang akurat dan menyesuaikan pengajaran mereka untuk memenuhi kebutuhan siswa [3].

Artificial Intelligence yang merupakan suatu sistem komputer yang dapat melakukan pekerjaan-pekerjaan atau tugas-tugas yang umumnya membutuhkan kepintaran manusia [4]. Berbagai upaya telah dilakukan untuk mendeteksi dan mencegah kecurangan dalam ujian daring, salah satunya adalah penggunaan teknologi kecerdasan buatan (AI) dan *machine learning*. Teknologi ini menawarkan potensi untuk mendeteksi pola-pola perilaku mencurigakan secara otomatis dan real-time, sehingga dapat memberikan solusi yang lebih efektif dibandingkan metode tradisional yang mengandalkan pengawasan manual. Deteksi potensi menyontek merupakan salah satu aplikasi AI yang dapat digunakan dalam *platform* ujian daring.

Pada penelitian sebelumnya yang dilakukan oleh Seng Zi Ong, Tee Connie, dan Michael Kah Ong Goh melakukan prediksi deteksi menyontek

dengan menggunakan model YOLOv5 dan mendapatkan akurasi sebesar 83% [5].

Pada penelitian yang dilakukan oleh Rehab Kareem Kadthim dan Zuhair Hussein Ali melakukan prediksi menyontek menggunakan dataset berupa nilai pelajar yaitu *quiz*, ujian tengah semester, dan juga ujian akhir. Penelitian ini menggunakan beberapa algoritma *machine learning* yaitu SVM, KNN, dan RF yang mendapatkan akurasi sebesar 96%, 95%, dan 84% [6].

Pada penelitian yang dilakukan oleh Waleed Alsabhan, penulis melakukan prediksi menyontek menggunakan dataset berupa informasi dari berbagai sumber, termasuk survei, data sensor, dan catatan institusi. Dataset ini berisi informasi tentang kehadiran siswa, kinerja akademik, dan perilaku sosial. Penelitian tersebut menggunakan algoritma LSTM yang mendapatkan akurasi sebesar 92% [7].

Pada penelitian sebelumnya yang dilakukan oleh Fairouz Hussein, Ayat Al-Ahmad, Subhieh El-Salhi, Esra'a Alshdaifat, dan Mo'taz Al-Hami menggunakan algoritma bernama CHEESE (*CHEating detection framework via multiple inStance learning*) yang mendapatkan hasil akurasi sebesar 87.58% dengan menggunakan dataset dari OEP dataset [8].

Pada penelitian sebelumnya, memang lumayan banyak yang sudah melakukan penelitian untuk mendeteksi menyontek. Akan tetapi, belum ada penelitian yang menggunakan algoritma *FeedForward Neural Network* dan juga MediaPipe Face Landmark.

Dengan memanfaatkan *multithreading* dan akselerasi GPU, MediaPipe memiliki kinerja yang sangat cepat dalam memproses volume data yang besar. Fitur yang ditawarkan oleh MediaPipe mencakup deteksi wajah, mesh wajah, iris, deteksi tangan, pose, holistik, segmentasi rambut, deteksi objek, pelacakan kotak, pelacakan gerakan instan, Objectron, dan KNIFT [9].

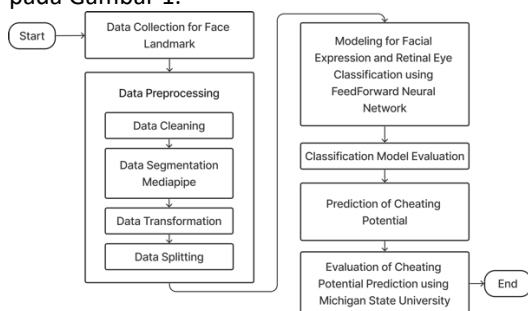
Feedforward Neural Network. *Feedforward neural network* merupakan alat yang sangat fleksibel untuk mengembangkan berbagai jenis model. Model ini terbentuk dari sejumlah besar

neuron yang saling terhubung dalam struktur yang rumit dan berlapis-lapis. *Neuron* dalam jenis jaringan saraf ini berinteraksi secara rumit, membentuk beberapa lapisan [10].

Oleh karena itu, tujuan dari penelitian ini adalah untuk membuat sebuah model deteksi potensi menyontek menggunakan algoritma *feedforward neural network* serta MediaPipe Face Landmark. Dengan pendekatan ini, diharapkan model pembelajaran dapat menghasilkan klasifikasi potensi menyontek dari dataset yang ada dengan akurasi yang memuaskan serta mampu mendeteksi adanya potensi menyontek atau tidak. Hasil penelitian ini diharapkan dapat menjadi referensi bagi pembaca serta pemerintah dalam upaya mengurangi praktik menyontek pada saat *online exam*.

2. METODE PENELITIAN

Pada penelitian ini, penulis menggunakan metode eksperimental. Metode eksperimental merupakan suatu pendekatan yang terlibat dalam pengumpulan data melalui percobaan yang diselenggarakan secara teratur dan terkendali [11]. Metode penelitian yang digunakan untuk mendeteksi potensi menyontek mengacu pada pendekatan berbasis gambar yang diubah menjadi titik koordinat face mesh. Penelitian ini menggunakan metode Feedforward neural network sebagai dasar algoritma untuk pembuatan model. Berikut merupakan gambar langkah-langkah penelitian yang bisa dilihat pada Gambar 1.



Gambar 1. Langkah-langkah penelitian

2.1 Data Collection

Data collection dilakukan dengan meminta orang-orang untuk membuat video yang berisikan *pose* yang sudah diinstruksikan oleh penulis. *Pose* yang dimaksud adalah netral, kepala menghadap ke atas, kepala menghadap kebawah, kepala menghadap ke kiri, kepala

menghadap ke kanan, bola mata melirik ke kiri, dan bola mata melirik ke kanan. Jumlah dataset yang terkumpul sebanyak 51 video yang diperoleh dari pengumpulan data yang dilakukan oleh penulis dengan cara meminta orang-orang membuat video tersebut. Dataset ini ditujukan hanya sebagai data yang digunakan untuk mengekstrak koordinat wajah menggunakan Mediapipe Face Landmark. Keputusan menggunakan *pose* tersebut sudah berdasarkan penelitian sebelumnya [12] [13]. *Pose* pada dataset ini juga berdasarkan dataset yang sudah dibuat oleh Sucianna Ghadati Rabiha pada Mendeley Data [14]. Pada akhir penelitian akan dilakukan pengujian terhadap algoritma yang digunakan pada penelitian ini menggunakan dataset dari Michigan State University [15]. Pada dataset ini terdapat 24 dataset video yang isinya 15 orang yang melakukan akting menyontek serta 9 orang yang melakukan ujian sebenarnya. Orang yang melakukan ujian sebenarnya tidak terlihat menyontek sehingga pengawasi mengambil tindakan untuk memancing agar orang tersebut melakukan aksi menyontek dengan cara berbicara, berjalan, memberikan buku kepada pelajar, dan sebagainya.

2.2 Pra-pemrosesan data

Data preprocessing atau pra-pemrosesan data adalah rangkaian langkah yang diambil untuk mempersiapkan data agar cocok dengan struktur yang dibutuhkan [16]. Berikut beberapa langkah *data preprocessing* pada penelitian ini.

2.2.1 Pembersihan Data

Data cleaning atau pembersihan data adalah langkah menghapus data yang tidak relevan atau tidak diperlukan [17]. Pada tahap *data cleaning*, langkah pertama yang dilakukan oleh penulis adalah mengubah dataset video menjadi gambar. Sebelum diubah menjadi data gambar, video dipotong per bagian sesuai dengan *pose* yang sudah di tetapkan dan disimpan kedalam *folder* sesuai dengan nama *class*. Setelah semua video sudah dipotong, video tersebut akan di konversi menjadi gambar. Hasil gambar yang tersimpan di *filter* satu per satu apakah gambar layak digunakan atau tidak. Gambar yang layak digunakan akan dipakai sebagai dataset yang akan dikonversi menjadi titik-titik koordinat. Sedangkan, gambar yang tidak layak akan dihapus dan tidak digunakan.

2.2.2 Data Segmentation

Data Segmentation adalah tahap penting untuk mengekstrak informasi dari data gambar yang telah diproses sebelumnya. Ini memungkinkan kuantifikasi dan identifikasi area yang sesuai dengan objek yang diminati dalam gambar secara otomatis [18].

Hasil dari gambar yang sudah dikonversi akan digunakan untuk mengubah gambar menjadi titik-titik koordinat menggunakan MediaPipe. MediaPipe memiliki total 478 titik koordinat. Titik-titik koordinat yang sudah didapatkan akan disimpan ke dalam *file* dengan ekstensi CSV. Pada file CSV tersebut akan berisi *class* dan titik koordinat. Total data yang dikonversi sebanyak 8809 gambar. Detail jumlah data tiap class yaitu netral sebanyak 1474, hadap_atas sebanyak 1377, hadap_bawah sebanyak 1187, hadap_kiri sebanyak 1177, hadap_kanan sebanyak 1126, retina_kiri sebanyak 1175 dan retina_kanan sebanyak 1293.

2.2.3 Transformasi Data

Data transformation atau transformasi data adalah langkah mengalihkan format data dari satu bentuk ke bentuk lainnya. *Data transformation* melibatkan tahapan pembersihan, perubahan, serta pengaturan ulang data agar dapat dipergunakan untuk keperluan yang beragam. Melalui transformasi data, berbagai sumber data dengan format yang beragam dapat digabungkan, dianalisis, dan dimodelkan sebagai satu kesatuan [19]. Pada tahap ini data *class* pada CSV akan di ubah menjadi numerik yaitu angka 0 untuk netral, angka 1 untuk hadap_atas, angka 2 untuk hadap_bawah, angka 3 untuk hadap_kanan, angka 4 untuk hadap_kiri, angka 5 untuk retina_kiri, dan angka 6 untuk retina_kanan.

2.2.4 Data Splitting

Data splitting merupakan metode dalam *machine learning* yang dipakai untuk memisahkan dataset menjadi dua bagian atau lebih. Saat melakukan *data splitting*, dataset akan dipisahkan menjadi *data training* dan *data testing* [20]. Rasio pembagian data di penelitian ini adalah sebanyak 80:20 yang artinya terdapat 80% *data training* dan 20% *data testing*.

2.3 Data Modeling

Data modeling dipahami sebagai teknik yang mencatat fitur-fitur elemen data dalam suatu

peta yang menggambarkan data yang digunakan dalam suatu proses. *Data modeling* menggambarkan bagaimana elemen data diorganisir dan berhubungan [21]. *Feature extraction* adalah proses yang mengubah dataset asli menjadi jumlah fitur yang lebih sedikit. Sebaliknya, pemilihan fitur menemukan elemen yang memberikan informasi paling banyak tentang masalah tersebut [22].

Dasar model dari penelitian ini menggunakan algoritma *Feedforward Neural Network* yang ada pada *library* TensorFlow Keras Sequential. *Feedforward Neural Network* ditujukan sebagai *feature extraction* sekaligus sebagai menghasilkan sebuah model yang akan digunakan untuk memberikan prediksi potensi tidak menyontek dan berpotensi menyontek.

2.3.1 Feedforward Neural Network

Feedforward neural network adalah satu dari banyak jaringan saraf yang paling awal dipakai dalam klasifikasi [23]. *Feedforward neural network* merupakan alat yang sangat fleksibel untuk mengembangkan berbagai jenis model, termasuk regresi non-linear, reduksi data, dan sistem dinamis non-linear. Model ini terbentuk dari sejumlah besar *neuron* yang saling terhubung dalam struktur yang rumit dan berlapis-lapis. *Neuron* dalam jenis jaringan saraf ini berinteraksi secara rumit, membentuk beberapa lapisan [10]. Persamaan (1) merupakan contoh penulisan persamaan untuk metode *feedforward neural network* yang mempunyai input sebesar p serta satu *hidden layer* dengan *neuron* sebesar m .

$$f(x_t, v, w) = g_2\left\{\sum_{j=0}^m v_j g_1\left[\sum_{i=0}^p w_{ji} x_{it}\right]\right\} \quad (1)$$

w adalah *weight vector* yang mengaitkan *input layer* dengan *hidden layer*, sementara v adalah *weight vector* yang mengaitkan *hidden layer* dengan *output layer*. *Activation layer* g_1 dan g_2 dipakai untuk mengubah transformasi di setiap *layer*. Selain dari itu, w_{ji} dan v_j adalah *weight* yang dihitung menggunakan proses *backpropagation* [23].

2.4 Model Evaluation

Model evaluation merupakan tahap yang melibatkan pengukuran kinerja model saat diterapkan pada dataset yang telah disediakan untuk pembelajaran [24]. Penulis memakai

metode *confusion matrix* sebagai metode *model evaluation*.

2.4.1 Confusion Matrix

Confusion matrix adalah sebuah metode yang digunakan sebagai alat pengukuran guna mengetahui jumlah keakuratan klasifikasi terhadap kelas menggunakan algoritma tertentu [25].

Tabel 1. Bentuk *Confusion matrix*

		Actual Values	
		Positive	Negative
Predicted Values	Positive	TP	FP
	Negative	FN	TN

True Positive atau TP adalah hasil prediksi *positive* yang sesuai dengan nilai sebenarnya. *True Negative* atau TP merupakan hasil prediksi *negative* yang sesuai dengan nilai sebenarnya. *False Positive* atau FP yaitu hasil prediksi *positive* yang tidak sesuai dengan nilai sebenarnya. *False Negative* atau FN merupakan hasil prediksi *negative* yang tidak sesuai dengan nilai sebenarnya. Dengan menggunakan *confusion matrix*, dapat ditentukan nilai akurasi, presisi, *recall* serta *f1-score*.

Accuracy adalah fraksi dari prediksi benar dibandingkan dengan jumlah prediksi yang dibuat oleh pengklasifikasi [26]. Persamaan (2) merupakan contoh penulisan persamaan untuk *accuracy*.

$$accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

Precision adalah jumlah kasus yang diperkirakan benar yang ternyata positif. Metrik ini menilai apakah model tersebut dapat dipercaya atau tidak [26]. Persamaan (3) merupakan contoh penulisan persamaan untuk *precision*.

$$precision = \frac{TP}{TP+FP} \quad (3)$$

Recall adalah jumlah kasus yang benar-benar positif yang dapat diprediksi dengan benar menggunakan model [26]. Persamaan (4) merupakan contoh penulisan persamaan untuk *recall*.

$$recall = \frac{TP}{TP+FN} \quad (4)$$

F1-score adalah gagasan gabungan tentang metrik *precision* dan *recall*. Ini berarti ketika

mencoba meningkatkan nilai presisi, *recall* akan turun dan sebaliknya [26]. Persamaan (5) merupakan contoh penulisan persamaan untuk *f1-score*.

$$F1_{score} = 2 \times \frac{Precision \times Recall}{Precision+Recall} \quad (5)$$

2.5 Prediksi Menyontek

Pada penelitian ini, penulis menggunakan algoritma sendiri dalam memprediksi menyontek yang akan dilakukan pengujian menggunakan dataset dari Michigan State University. Algoritma ini digunakan untuk mendeteksi apakah pelajar berpotensi menyontek atau tidak berpotensi menyontek.

Untuk menentukan prediksi berpotensi menyontek menggunakan *class* *hadap_atas*, *hadap_bawah*, *hadap_kiri* dan *hadap_kanan* yang dijumlahkan setiap 5 detik. Jika hasil penjumlahan tersebut lebih besar dari hasil maksimal yang didapatkan dari *class* *netral*, *retina_kiri* dan *retina_kanan* maka akan menghasilkan output berpotensi menyontek jika tidak lebih besar maka akan menghasilkan *output* tidak berpotensi menyontek. Persamaan ini sudah di validasi oleh pakar serta sudah diuji menggunakan dataset dari Michigan State University yang ada pada hasil dan pembahasan. Berikut merupakan hasil persamaan yang dapat dilihat pada Persamaan (6), Persamaan (7), dan Persamaan (8).

$$p = \sum_{n=1}^5 hki_n + hka_n + ha_n + hb_n \quad (6)$$

$$q = \max(\sum_{n=1}^5 ne_n, \sum_{n=1}^5 rki_n, \sum_{n=1}^5 rk_n) \quad (7)$$

$$p > q \rightarrow Berpotensi Menyontek \quad (8)$$

n merupakan durasi prediksi setiap 5 detik. *p* merupakan hasil jumlah *frame* terdeteksi *hadap_kiri*, *hadap_kanan*, *hadap_atas*, dan *hadap_bawah* selama 5 detik. *hki_n* merupakan jumlah *frame* terdeteksi wajah *hadap_kiri*. *hka_n* merupakan jumlah *frame* terdeteksi wajah *hadap_kanan*. *ha_n* merupakan jumlah *frame* terdeteksi wajah *hadap_atas*. *hb_n* merupakan jumlah *frame* terdeteksi wajah *hadap_bawah*. *q* merupakan hasil maksimal yang didapatkan dari jumlah *frame* terdeteksi *netral*, *retina_kiri* dan *retina_kanan*. *ne_n* merupakan jumlah *frame* terdeteksi wajah *netral*. *rki_n* merupakan jumlah *frame* terdeteksi *retina_kiri*. *rk_n* merupakan jumlah *frame* terdeteksi *retina_kanan*.

3. HASIL DAN PEMBAHASAN

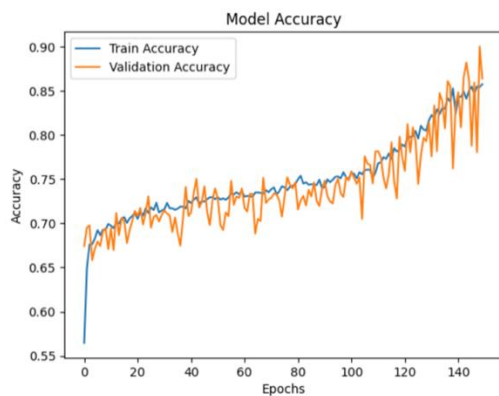
3.1 Data Preprocessing

Dalam lingkup studi ini, penulis sudah mendapatkan total 51 video dataset. Video dataset tersebut dipotong sesuai pose yang sudah ditentukan agar memudahkan dalam konversi video menjadi gambar. Setelah sudah di konversi menjadi gambar, didapatkan total gambar sebesar 9287 gambar. Gambar yang didapatkan akan dilakukan *data segmentation* menggunakan MediaPipe Face Landmark untuk mendapatkan titik-titik koordinat. Didapatkan total 8809 baris di file CSV yang menandakan 478 file gambar yang tidak terdeteksi dikarenakan gambar yang blur.

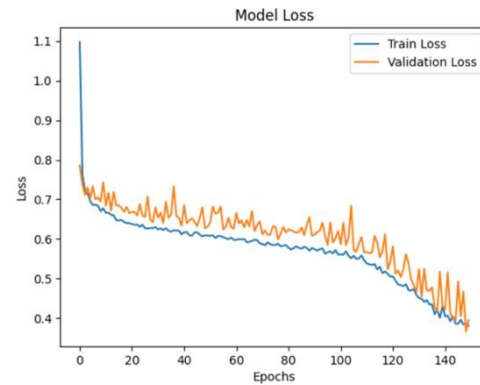
Penelitian ini terdapat 7 *class*, yaitu netral, hadap_atas, hadap_bawah, hadap_kanan, hadap_kiri, retina_kanan, dan retina_kiri. *Class* tersebut akan diubah menjadi bentuk numerik yaitu angka 0 untuk netral, angka 1 untuk hadap_atas, angka 2 untuk hadap_bawah, angka 3 untuk hadap_kanan, angka 4 untuk hadap_kiri, angka 5 untuk retina_kiri, dan angka 6 untuk retina_kanan. Setelah melakukan *data transformation*, dataset akan di bagi dengan ratio 80:20 yang artinya terdapat 80% *data training* dan 20% *data testing*.

3.2 Modeling

Penulis menggunakan metode *Feedforward Neural Network* dengan 2 *hidden layer*. Algoritma ini terdapat pada *library* TensorFlow Keras. Pelatihan ini menggunakan *optimizer* Adam, *epochs* sebesar 150, dan *batch_size* sebesar 8. Pada akhir pelatihan model didapatkan *accuracy* sebesar 85% dan *val_accuracy* sebesar 86%. Berikut merupakan grafik dari *model accuracy* yang dapat dilihat pada Gambar 2 dan *model loss* yang dapat dilihat pada Gambar 3.



Gambar 2. Grafik Model Accuracy



Gambar 3. Grafik Model Loss

3.3 Model Evaluation

Pada *model evaluation*, penulis menggunakan metode *confusion matrix* untuk menghitung *accuracy*, *precision*, *recall*, dan *f1-score* tiap *class*. Pada Gambar 4 terlihat *class* yang memiliki persentase *precision*, *recall*, dan *f1-score* tertinggi adalah *class* hadap_kanan dan hadap_kiri dengan persentase 100%. Persentase *accuracy* keseluruhan model sebesar 86%.

Pada grafik *model accuracy* yang terdapat pada Gambar 2 dan model loss pada Gambar 3, terlihat bahwa grafik tersebut menunjukkan hasil yang fluktuatif. Hal ini disebabkan oleh dataset yang berupa koordinat serta terdapat kemiripan antara *class* netral, retina_kiri dan retina_kanan. Hal ini terbukti dengan *f1-score* yang dapat dilihat pada Gambar 4. Penyebab dari nilai *f1-score* yang cukup rendah pada *class* netral, retina_kiri dan retina_kanan adalah kemiripan pada posisi wajah, perbedaan hanya terletak pada arah retina mata saja.

Confusion Matrix:

```

[[252  9  11  0  0  0  43]
 [ 24 255  0  1  0  1  11]
 [  7  0 207  0  0  0  23]
 [  0  0  0 221  0  0  0]
 [  0  0  0  0 225  0  0]
 [100  5  1  0  0  122  0]
 [  3  0  1  0  0  0 240]]

```

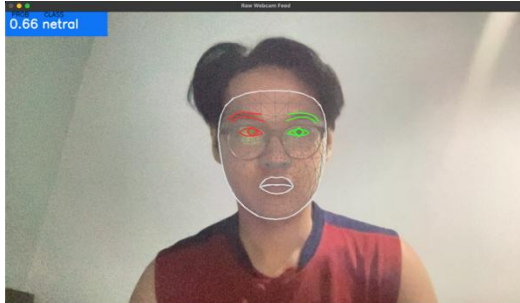
Classification Report:

	precision	recall	f1-score	support
netral	0.65	0.80	0.72	315
hadap_atas	0.95	0.87	0.91	292
hadap_bawah	0.94	0.87	0.91	237
hadap_kanan	1.00	1.00	1.00	221
hadap_kiri	1.00	1.00	1.00	225
retina_kiri	0.99	0.54	0.70	228
retina_kanan	0.76	0.98	0.86	244
accuracy			0.86	1762
macro avg	0.90	0.87	0.87	1762
weighted avg	0.89	0.86	0.86	1762

Gambar 4. Hasil evaluasi model

3.4 Testing

Setelah tahap *model evaluation* selesai, penulis melanjutkan dengan melakukan testing secara langsung menggunakan *raw camera*. Gambar 5 merupakan salah satu *class* dari hasil testing yang dilakukan.



Gambar 5. Hasil *testing* pada *raw camera*

Setelah melakukan testing pada *raw camera*, untuk melakukan deteksi berpotensi menyontek atau tidak berpotensi menyontek, penulis mengambil rata-rata *class* dari *frame* yang terdeteksi setiap 5 detik. Setiap *frame* dihitung sesuai logika yang dibuat. Jika *class* *hadap_kiri*, *hadap_kanan*, *hadap_atas*, dan *hadap_bawah* mendapatkan total nilai rata-rata yang lebih tinggi dari *class* lainnya maka akan terdeteksi "Berpotensi Menyontek". Jika *class* *netral* mendapatkan total nilai rata-rata yang paling tinggi dibanding dengan *class* lainnya maka akan terdeteksi "Tidak Berpotensi Menyontek". Jika *class* *netral*, *retina_kiri*, dan *retina_kanan* mendapat total nilai rata-rata yang lebih besar dari *class* lainnya maka akan terdeteksi "Tidak Berpotensi Menyontek". Gambar 6 merupakan hasil testing deteksi menyontek dan Gambar 7 merupakan pseudocode.

```
Tidak Berpotensi Menyontek, Sum of Labels: 138
total hadap_kiri, hadap_kanan, hadap_atas, hadap_bawah: 0
total netral: 138
total retina_kiri, retina_kanan, netral: 138
Berpotensi Menyontek, Sum of Labels: 139
total hadap_kiri, hadap_kanan, hadap_atas, hadap_bawah: 104
total netral: 34
total retina_kiri, retina_kanan, netral: 35
```

Gambar 6. Hasil deteksi menyontek

```
Initialize label_counts as an array of zeros with length 7
If elapsed_time is greater than or equal to 5.0:
  If predictions is not empty:
    Calculate the sum of label_counts for 'hadap_kiri', 'hadap_kanan', 'hadap_atas', and
    'hadap_bawah'
    If the sum is greater than the maximum of label_counts for other classes:
      Set classification_result to "Berpotensi Menyontek"
    Else if the count of 'netral' is greater than the maximum count of other classes:
      Set classification_result to "Tidak Berpotensi Menyontek"
    Else if the sum of 'retina_kiri', 'retina_kanan', and 'netral' is greater than or equal to
    the average of label_counts:
      Set classification_result to "Tidak Berpotensi Menyontek"
    Else:
      Set classification_result to "Berpotensi Menyontek"
  Set result_text to include classification_result and the sum of label_counts
  Display result_text
  Draw result_text on the usage
  Reset start_time, predictions, and label_counts for the next time window
Else:
  Display "No predictions made."
```

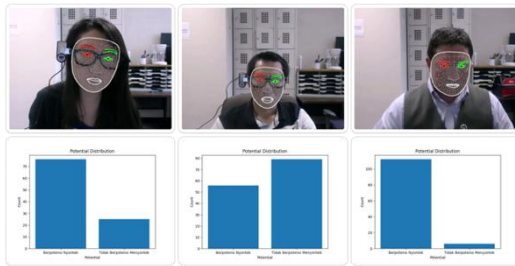
Gambar 7. Pseudocode

Pertimbangan dalam penggunaan algoritma yang digunakan untuk menentukan klasifikasi antara berpotensi tidak menyontek serta berpotensi menyontek yang terdapat pada Gambar 7 adalah ekspresi wajah netral. Wajah netral merupakan pandangan fokus ke depan yang menandakan siswa melihat ke layar. Kemudian *class* *retina_kiri* dan *retina_kanan* merupakan *class* yang digunakan sebagai pertimbangan pelajar membaca soal. *Class* *hadap_atas*, *hadap_bawah*, *hadap_kiri*, dan *hadap_kanan* digunakan sebagai *variable* yang menandakan adanya indikasi kecurangan karena terdapat distraksi sehingga pelajar melihat arah lain selain layar.

Sebagai pembuktian dalam penggunaan algoritma yang dibuat pada penelitian ini, penulis menggunakan dataset dari Michigan State University. Dari 24 video dataset berlabel menyontek ini, didapatkan 22 dari 24 video dinyatakan melakukan aksi menyontek. Sehingga dapat disimpulkan akurasi dari prediksi potensi menyontek pada penelitian ini sebesar 91.6% yang dapat dilihat pada Persamaan (9).

$$accuracy = \frac{22}{24} \times 100\% = 91.6\% \quad (9)$$

Pada Gambar 8, terdapat 3 contoh hasil pengujian menggunakan dataset dari Michigan State University. Pada grafik, sumbu x yang berlabel *potential* berisikan berpotensi menyontek dan tidak berpotensi menyontek. Sedangkan, sumbu y yang berlabel *count* merupakan jumlah dari hasil prediksi tiap 5 detik. Tiap video memiliki durasi yang berbeda-beda sehingga memiliki jumlah total yang berbeda. Gambar pertama merupakan hasil prediksi video ketika melakukan ujian sebenarnya yang mendapatkan hasil prediksi berpotensi menyontek. Gambar kedua merupakan hasil prediksi video ketika melakukan akting menyontek yang mendapatkan hasil prediksi tidak berpotensi menyontek. Gambar ketiga merupakan hasil prediksi video ketika melakukan akting menyontek yang mendapatkan hasil prediksi berpotensi menyontek. Berikut merupakan beberapa gambar hasil pengujian yang terdapat pada Gambar 8.



Gambar 8. Hasil pengujian

4. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan oleh penulis, dapat disimpulkan bahwa penggunaan MediaPipe Face Landmark untuk mendapatkan titik-titik koordinat wajah serta algoritma *feedforward neural network* cukup memuaskan karena mendapatkan nilai akurasi prediksi potensi menyontek sebesar 91.6% yang didapatkan dari hasil pengujian menggunakan dataset dari Michigan State University. Dari 24 video dataset menyontek didapatkan 22 video terdeteksi menyontek. Sistem ini cukup efektif diimplementasikan dalam sistem deteksi potensi menyontek saat melakukan ujian daring.

DAFTAR PUSTAKA

- [1] W. W. Muhlshottin and M. V. Roesminingsih, "Pelaksanaan Fungsi Fungsi Manajemen Kelompok Bermain RA Kartini Desa Trutup Kecamatan Plumpang Kabupaten Tuban," *Jurnal Pendidikan Untuk Semua*, vol. 4, no. 1, pp. 116-123, 2020.
- [2] A. Anitasari, O. Pandansari, R. Susanti, K. Kurniawati and A. Aziz, "Pengaruh Efikasi Diri terhadap Perilaku Menyontek Siswa Sekolah Dasar selama Pembelajaran Daring," *Jurnal Penelitian Ilmu Pendidikan*, vol. 14, no. 1, pp. 82-90, 2021.
- [3] M. Undorf, E. Y. Mah, D.-L. L. McDonald, Z. I. Hamzagic, R. Burnell, M. Garry and D. M. Bernstein, "People who cheat on tests accurately predict their performance on future tests," *Learning and Instruction*, vol. 66, no. 101295, 2020.
- [4] M. S. Y. Lubis, "IMPLEMENTASI ARTIFICIAL INTELLIGENCE PADA SYSTEM MANUFAKTUR TERPADU," *SEMNASTEK UISU 2021*, vol. 4, no. 1, pp. 1-7, 2021.
- [5] S. Z. Ong, T. Connie and M. K. O. Goh, "Cheating Detection for Online Examination Using Clustering Based Approach," *JOIV International Journal on Informatics Visualization*, vol. 7, no. 3-2, p. 2075, 2023.
- [6] R. K. Kadthim and Z. H. Ali, "Cheating Detection in online exams using machine learning," *Journal Of AL-Turath University College*, vol. 2, no. 35, pp. 35-41, 2023.
- [7] W. Alsabhan, "Student Cheating Detection in Higher Education by Implementing Machine Learning and LSTM Techniques," *Sensors*, vol. 23, no. 8, p. 4149, 2023.
- [8] Y. Liu, J. Ren, J. Xu, X. Bai, R. Kaur and F. Xia, "Multiple Instance Learning for Cheating Detection and Localization in Online Examinations," *IEEE Transactions on Cognitive and Developmental Systems*, pp. 1-12, 2024.
- [9] S. N. Budiman, S. Lestanti, S. M. Evvandri and R. K. Putri, "FINGER GESTURE RECOGNITION TO CONTROL VOLUME ON COMPUTER USING OPENCV AND MEDIAPIPE LIBRARY," *ANTIVIRUS: Jurnal Ilmiah Teknik Informatika*, vol. 16, no. 2, pp. 223-232, 2022.
- [10] P. D. Saputri and P. P. Oktaviana, "Comparison of Feedforward Neural Network and Classical Statistics Methods: Application in Finance Authors," *Jurnal Matematika, Statistika dan Komputasi*, vol. 19, no. 3, pp. 537-548, 2023.
- [11] A. S. A. Anshari, M. A. Ihsan, R. A. Duyo and H. Nirwana, "ANALISA TENAGA SURYA TERKONSENTRASI PADA KONVERSI CAHAYA SINAR MATAHARI MENJADI ENERGI LISTRIK TERHADAP SUPLAI AIR," *Vertex Elektro*, vol. 15, no. 2, pp. 11-18, 2023.
- [12] M. M. Masud, K. Hayawi, S. S. Mathew, T. Michael and M. E. Barachi, "Smart Online Exam Proctoring Assist for Cheating Detection," *Advanced Data Mining and Applications*, vol. 13087, p. 118-132, 2022.
- [13] A. Zaffar, M. Jawad and M. Shabbir, "A Novel CNN-RNN Model for E-Cheating Detection Based on Video Surveillance,"

- University of Wah Journal of Computer Science*, vol. 5, pp. 1-13, 2023.
- [14] S. G. Rabiha, *Cheating Scenario Dataset in Online Exam*, Mendeley Data, 2024.
- [15] Y. Atoum, L. Chen, A. X. Liu and S. D. H. Hsu, "Automated Online Exam Proctoring," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1609 - 1624, 2017.
- [16] N. K. Pratama, "DETEKSI LAMPU LALU LINTAS DAN ZEBRA CROSS MENGGUNAKAN MOBILENETV2 SINGLE SHOT DETECTOR,," *Jurnal Informatika Kaputama (JIK)*, vol. 7, no. 2, pp. 225-232, 2023.
- [17] R. D. Romadhona, "emanfaatan Teknik Clustering untuk Mencapai Kecerdasan Bisnis Menggunakan Algoritma K-Means," *Seminar Nasional Sistem Informasi (SENASIF)*, vol. 6, no. 1, pp. 3015-2023, 2022.
- [18] S. Nawoya, F. Ssemakula, R. Akol, Q. Geissmann, H. Karstoft, K. Bjerger, C. Mwirize, A. Katumba and G. Gebreyesus, "Computer vision and deep learning in insects for food and feed production: A review," *Computers and Electronics in Agriculture*, vol. 216, p. 108503, 2024.
- [19] E. R. Fitria and F. Rozci, "Penerapan Metode Regresi Least Absolute Shrinkage and Selection Operator (LASSO) dan Regresi Linier untuk Memprediksi Tingkat Kemiskinan di Indonesia," *Jurnal Ilmiah Sosio Agribis (JISA)*, vol. 22, no. 2, pp. 123-132, 2022.
- [20] H. Santoso, R. A. Putri and S. Sahbandi, "Deteksi Komentar Cyberbullying pada Media Sosial Instagram Menggunakan Algoritma Random Forest," *Jurnal Manajemen informatika (JAMIKA)*, vol. 13, no. 1, pp. 62-72, 2023.
- [21] D. Martinez-Mosquera, R. Navarrete and S. Lujan-Mora, "Modeling and Management Big Data in Databases—A Systematic Literature Review," *Sustainability*, vol. 12, no. 2, p. 634, 2020.
- [22] R. E. Nogales and M. E. Benalcázar, "Analysis and Evaluation of Feature Selection and Feature Extraction Methods," *International Journal of Computational Intelligence Systems*, vol. 16, no. 153, pp. 1-13, 2023.
- [23] T. P. Tan, C. K. Lim and W. R. E. A. Rahman, "Sliding Window and Parallel LSTM with Attention and CNN for Sentence Alignment on Low-Resource Languages," *Pertanika Journal Science & Technology*, vol. 30, no. 1, pp. 97-121, 2022.
- [24] L. Priyambodo, H. L. Fuadi, N. Nazhifah, I. Huzaimi, A. B. Prawira, T. E. Saputri, M. A. Afandi, E. S. Nugraha, A. Wicaksono and P. K. Goran, "Klasifikasi Kematangan Tanaman Hidroponik Pakcoy Menggunakan Metode SVM," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 6, no. 1, pp. 153-160, 2022.
- [25] L. Qadrini, A. Seppewali and A. Aina, "Decision Tree dan Adaboost pada Klasifikasi Penerima Program Bantuan Sosial," *Jurnal Inovasi Penelitian*, vol. 2, no. 7, pp. 1959-1966, 2021.
- [26] A. Tasnim, M. Saiduzzaman, M. A. Rahman, J. Akhter and A. S. M. M. Rahaman, "Performance Evaluation of Multiple Classifiers for Predicting Fake News," *Journal of Computer and Communications*, vol. 10, no. 9, pp. 1-21, 2022.
- [27] R. K. Kadhim and Z. H. Ali, "Cheating Detection in online exams using machine learning," *Journal Of AL-Turath University College*, vol. 2, no. 35, pp. 35-41, 2023.