

## PERBANDINGAN KINERJA INCEPTION-RESNETV2, XCEPTION, INCEPTION-V3, DAN RESNET50 PADA GAMBAR BENTUK WAJAH

Fitriana Masruroh<sup>\*1</sup>, Bayu Surarso<sup>2</sup>, Budi Warsito<sup>3</sup>

<sup>1,2,3</sup>Universitas Diponegoro, Semarang

Email: <sup>1</sup>fitrianamasruroh@students.undip.ac.id, <sup>2</sup> bayuurarso@yahoo.com

<sup>\*</sup>Penulis Korespondensi

(Naskah masuk: 21 April 2021, diterima untuk diterbitkan: 27 Februari 2023)

### Abstrak

Saat ini, klasifikasi bentuk wajah banyak diterapkan dalam berbagai bidang. Dalam bidang industri fashion dapat digunakan untuk pemilihan gaya rambut, pemilihan bingkai kaca mata, tata rias, dan mode lainnya. Selain itu, dalam bidang medis bentuk wajah digunakan untuk bedah plastik. Identifikasi bentuk wajah adalah tugas yang menantang karena kompleksitas wajah, ukuran, pencahayaan, usia dan ekspresi. Banyak metode yang dikembangkan untuk memberikan hasil akurasi terbaik dalam klasifikasi bentuk wajah. Deep learning menjadi tren dibidang komputer vision karena memberikan hasil yang paling baik dari pada metode sebelumnya. Makalah ini mencoba menyajikan perbandingan kinerja klasifikasi wajah dengan empat arsitektur deep learning Xception, ResNet50, InceptionResNet-v2, Inception-v3. *Dataset* yang digunakan berjumlah 4500 gambar yang terbagi lima kelas *heart, long, oblong, square, round*. Berbagai pengoptimal *deep learning* diantaranya; *transfer learning, optimizer deep learning, dropout* dan fungsi aktivasi diterapkan untuk meningkatkan kinerja model. Perbandingan antara berbagai model CNN didasarkan kinerja metrik seperti *accuracy, recall, precision* dan *F1-score*. Dengan demikian dapat disimpulkan bahwa model Inception-ResNet-V2 menggunakan fungsi aktivasi Mish dan *optimizer Nadam* mencapai nilai tertinggi dengan *accuracy* dan *f1-score* masing-masing 92.00%, dan penggunaan waktu 65.0 menit.

**Kata kunci:** *Bentuk wajah, deep learning, Xception, ResNet50, InceptionV3, InceptionResNetV2, optimizer,*

## PERFORMANCE COMPARISON OF INCEPTION-RESNETV2, XCEPTION, INCEPTION-V3, AND RESNET50 ON FACE SHAPE IMAGES

### Abstract

Currently, face shape classification is widely applied in various fields. In the fashion industry, it can be used for hairstyle selection, eyeglass frame selection, makeup, and other modes. In the medical field, the face shape is used for plastic surgery. Identification of face shape is a challenging task due to the complexity of the face, size, lighting, age and expression. Many methods have been developed to provide the best accuracy results in the classification of face shapes. Deep learning is becoming a trend in the field of computer vision because it gives the best results than the previous method. This paper attempts to present a comparison of the performance of face classification with four deep learning architectures Xception, ResNet50, InceptionResNet-v2, Inception-v3. The dataset used is 4500 images divided into five classes *heart, long, oblong, square, round*. Various deep learning optimizers include; *transfer learning, deep learning optimizer, dropout* and activation functions are implemented to improve model performance. Comparisons between various CNN models are based on performance metrics such as *accuracy, recall, precision* and *F1-score*. Thus, it can be concluded that the Inception-ResNet-V2 model using the Mish activation function and the Nadam optimizer achieves the highest value with an accuracy and f1-score of 92.00%, and a time usage of 65.0 minutes. Thus, it can be concluded that the Inception-ResNet-V2 model using the Mish activation function and the Nadam optimizer achieves the highest value with an accuracy and f1-score of 92.00%, and a time usage of 65.0 minutes.

**Keywords:** *Face shape, deep learning, Xception, ResNet50, InceptionV3, InceptionResNetV2, optimizer*

### 1. PENDAHULUAN

Wajah merupakan rangsangan visual penting dalam kehidupan sehari-hari. Variasi antarindividu

dalam bentuk wajah merupakan salah satu fenotipe paling unik yang terlihat pada manusia sebagai identifikasi setiap individu. Pengetahuan tentang bentuk wajah telah dipelajari secara menyeluruh

dalam bidang psikologi sosial (Liu *dkk.*, 2017), dan sudah banyak diterapkan dalam berbagai bidang. Misalnya dalam bidang industri fashion dapat digunakan untuk menentukan gaya rambut (Pasupa, Sunhem, & Loo, 2019), pemilihan bingkai kaca mata (Zafar dan Popa, 2016), tata rias, perhiasan dan mode lainnya. Dalam bidang medis untuk bedah plastik (Bouguila dan Khohtali, 2020). Meskipun identifikasi bentuk wajah telah diterapkan berbagai bidang, tetapi masih menjadi masalah yang sangat menantang dalam menentukan metode terbaik untuk memperoleh akurasi tinggi, karena kompleksitas wajah, ukuran, pencahayaan, usia, dan ekspresi.

Dalam literatur, banyak sistem klasifikasi bentuk wajah dipublikasikan dari metode mengekstrak fitur wajah secara manual kemudian meneruskan ke tiga pengklasifikasi untuk klasifikasi termasuk *Linear Discriminant Analysis* (LDA), *Artificial Neural Network* (ANN), dan *Support Vector Machine* (SVM) (Sunhem dan Pasupa, 2016), metode SVM (Sarakan, Charoenpong dan Charoensiriwath, 2014), dan *k-Nearest Neighbours* (KNN) (Zafar and Popa, 2016). Hasil akurasi secara keseluruhan diperoleh pada kisaran 64.2% - 73.68%. Selain itu pendekatan *deep learning* seperti *Convolutional Neural Network* (CNN) telah digunakan dalam klasifikasi bentuk wajah. Pasupa, Sunhem, & Loo, (2019) mempresentasikan pendekatan *hybrid* yang menggabungkan *Convolutional Neural Network* (CNN) VGG dengan SVM untuk klasifikasi bentuk wajah. Dataset yang digunakan berjumlah 500 gambar yang terdiri dari lima kelas. Dari hasil penelitian, peneliti ini mendapatkan akurasi terbaik yaitu sebesar 70.3% lebih baik dari pada menggunakan fitur secara individual. Alzahrani, Al-Nuaimy, & Al-Bander (2019) menggabungkan CNN InceptionV3 dengan *histogram of oriented gradients* (HOG) dan *facial landmarks*. Penelitian ini menggunakan 500 dataset gambar yang terbagi menjadi 5 kelas bentuk wajah yaitu *heart*, *long*, *square*, *oval*, dan *round*. Hasil penelitian menunjukkan penggabungan metode InceptionV3, HOG, dan *facial landmarks* mencapai akurasi tertinggi sebesar 81.1%. Tio (2019) mengadopsi CNN InceptionV3 yang telah dilatih (*pre-trained*) untuk mengklasifikasikan bentuk wajah menggunakan fitur yang diekstraksi secara otomatis oleh CNN. Dataset yang digunakan sebanyak 500 gambar. Hasilnya menunjukkan bahwa model InceptionV3 *pre-trained* dapat melakukan klasifikasi bentuk wajah dengan baik dan mengungguli metode klasifikasi dengan teknik *Machine Learning* (ML).

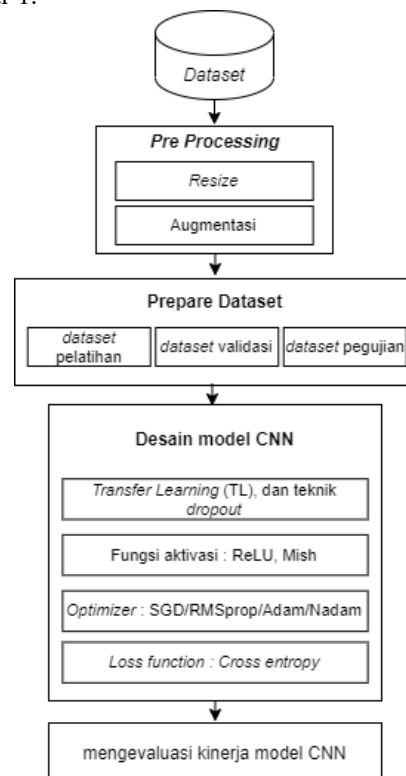
Penelitian dalam *Deep Learning* (DL) telah berkembang dengan memperkenalkan berbagai arsitektur untuk mencapai hasil yang luar biasa. Mengusulkan algoritme optimasi untuk menemukan nilai optimal dari *hyperparameter* arsitektur DL, Selain itu, meningkatkan inisialisasi bobot dengan *transfer learning*, *dropout* sebagai metode *regularization* untuk mengurangi *overfitting* selama

proses pengujian model. Oleh karena itu berbagai pengoptimalan pada DL dikembangkan peneliti untuk meningkatkan kinerja model sehingga mencapai hasil yang lebih baik dalam klasifikasi gambar. Misra (2019) akurasi yang dicapai dengan menggunakan fungsi aktivasi Mish memberikan peningkatan kinerja 1% - 3% dari model menggunakan *Rectified Linear Unit* (ReLU) pada *dataset* CIFAR 10 (Krizhevsky dan Hinton, 2009).

Pengembangan pendekatan *deep learning* memberikan hasil terbaik dibandingkan dengan metode tradisional berdasarkan akurasi (Tio, 2019). Sehingga mendorong peneliti untuk mengkaji pendekatan *deep learning* dalam klasifikasi bentuk wajah. Penelitian ini mencoba menyajikan analisis perbandingan kinerja untuk melakukan klasifikasi bentuk wajah dengan empat arsitektur *deep learning* Xception, ResNet50, InceptionResNet-v2, dan Inception-v3 yang sebelumnya digunakan untuk tugas pengenalan gambar. Berbagai *optimizer* DL diantaranya; RMSprop, SGD, Adam dan Nadam juga diterapkan dalam penelitian ini. Selanjutnya, menerapkan fungsi ReLU dan Fungsi aktivasi Mish yang dianggap menjadi fungsi aktivasi yang efektif. Untuk evaluasi yang komprehensif dengan membandingkan *accuracy*, *precision*, *recall* dan *F1-score*.

## 2. METODE PENELITIAN

Metodologi penelitian yang dilakukan terbagi menjadi lima bagian. Seperti yang ditunjukkan pada gambar 1.



Gambar 1. Alur metodologi penelitian

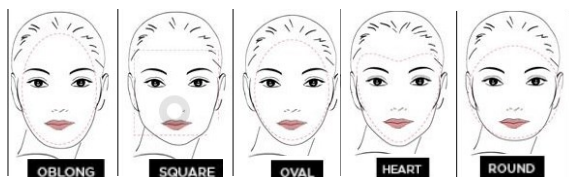
Bagian pertama *pra-processing* ini mencakup dua langkah yaitu; *resize* dan *augmentasi*. Bagian kedua menyiapkan *dataset* bentuk wajah untuk meningkatkan model CNN dan dibagi menjadi tiga set data, yang berisi *dataset* pelatihan, validasi dan pengujian. Bagian ketiga membangun model CNN dengan *transfer learning* dengan melibatkan teknik *dropout* dan fungsi aktivasi. Bagian keempat menggunakan fungsi *cross-entropy* dan *optimizer deep learning* termasuk SGD, RMSProp, Adam, dan Nadam, untuk menguji kinerja model CNN dalam klasifikasi bentuk wajah. Bagian kelima mengevaluasi performa klasifikasi menggunakan data validasi dan pengujian. Hasil pengujian ditampilkan dengan *confusion matrix* dan diagram *Receiver Operating Characteristic* (ROC) dan *Area Under Curve* (AUC).

### 2.1. Pengumpulan dataset

*Dataset* yang digunakan dalam penelitian ini adalah *dataset* bentuk wajah (Face Shape Dataset | Kaggle), *dataset* wajah manusia (Grgic and Delac, 2020). *Dataset* yang digunakan berisi 500 gambar per bentuk wajah atau 4500 gambar secara keseluruhan.

Sebelum gambar dilakukan proses pelatihan dan pengujian dilakukan terlebih dahulu *pre-processing image* yaitu mengubah ukuran gambar menjadi 224 x 224 x 3 dan 229 x 229 x 3. Normalisasi juga dipertimbangkan dengan membagi nilai pixel dengan 225 agar sesuai dengan nilai awal model. *Augmentasi* gambar digunakan untuk menambah variasi dari *dataset* dan mengurangi masalah *overfitting*.

*Dataset* dibagi menjadi tiga bagian yaitu *dataset* pengujian, validasi dan pengujian, yang masing-masing 70%, 20 % dan 10%. Gambar 2 menunjukkan contoh gambar *dataset* bentuk wajah dari masing-masing kelas.



Gambar 2. Contoh bentuk wajah

### 2.2. Arsitektur Deep Learning

Setelah pengembangan arsitektur AlexNet (Hinton et al., 2012), CNN menjadi semakin populer dalam *computer vision*. Periode revolusioner arsitektur CNN yang canggih dimulai untuk klasifikasi gambar. Dalam artikel ini kami mempertimbangkan empat model CNN yang populer dan sukses seperti; ResNet50 (He et al., 2016), Inception v3 (Szegedy et al., 2016), Inception-ResNet v2 (Szegedy et al., 2017) dan Xception (Chollet, 2017) untuk klasifikasi bentuk wajah. Tabel 1 menunjukkan detail spesifikasi arsitektur CNN yang diusulkan pada penelitian ini.

Tabel 1. Detail spesifikasi arsitektur CNN

Arsitektur CNN	Input	Parameter	Depth
Xception	229 x 229 x 3	22,910,480	126
Inception-ResNet V2	229 x 229 x 3	55,873,736	572
ResNet50	224 x 224 x 3	25,636,712	50
Inception V3	229 x 229 x 3	23,851,784	159

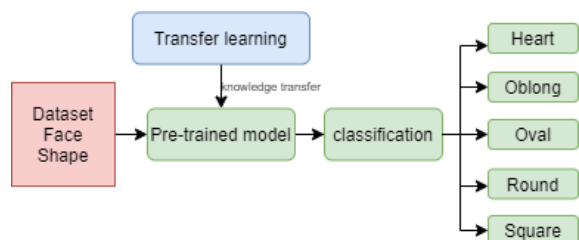
*SoftMax* (He dkk., 2018) diterapkan pada lapisan terakhir *neural network* untuk memberikan *output* pada klasifikasi *multiclass*. *SoftMax* meminta *output neural network* untuk merepresentasikan probabilitas bahwa *input* jatuh dalam setiap kelas. Persamaan (1) menunjukkan fungsi *SoftMax* dengan *input*  $x_i$ , jumlah kelas  $n$  dan kelas ke- $j$ .

$$\text{SoftMax}(x_i) = \frac{\exp(x_i)}{\sum_j^n \exp(x_j)} \quad (1)$$

### 2.3. Metode optimasi

#### Transfer learning

*Transfer learning* merupakan metode atau teknik yang menggunakan model yang sudah dilatih sebelumnya pada permasalahan yang serupa. Teknik ini memungkinkan penggunaan model terlatih secara langsung sebagai *pra-processing* ekstraksi fitur dan di integrasi kedalam model yang sepenuhnya baru. *Transfer learning* telah diterapkan pada *face recognition* didasarkan pada penelitian R. M. Prakash (2019) dan Tio (2019). Gambar 3 menunjukkan proses *transfer learning* dengan metode CNN menggunakan bobot yang sudah dilatih sebelumnya.



Gambar 3. CNN dengan transfer learning

#### Dropout

*Dropout* adalah regulator baru yang diusulkan untuk mencegah *overfitting* pada *feedforward neural networks* (Hinton dkk., 2012). *Overfitting* merupakan kondisi dimana semua data yang telah melalui proses pelatihan dan mencapai akurasi dengan persentase yang baik, tetapi terjadi ketidaksesuaian pada proses prediksi. *Dropout* pada struktur model adalah dengan mengabaikan beberapa *neuron* acak dalam model CNN, sehingga *neuron* yang diabaikan tidak akan mengirimkan data ke *neuron* lain. *Dropout* 0.5 pada *fully connected layer* akan mengabaikan 50% *neuron* di lapisan itu (Srivastava dkk., 2014).

### 2.4. Optimizer Deep Learning

*Gradient descent* adalah salah satu algoritme paling populer untuk mengoptimalkan pada *neural network*. *Gradient descent* merupakan cara untuk meminimalkan *objective function*  $J(\theta)$  dengan parameter model  $\theta \in \mathbb{R}$  dengan memperbaharui

parameter dalam arah berlawanan dari *gradient objective function*  $\nabla J(\theta)$  w.r.t ke parameter (Ruder, 2016).

Dalam penelitian ini kami menggunakan empat *optimizer* DL yang canggih untuk melatih model DL yang mencapai akurasi validasi tertinggi dan F1-score pada tingkat tertinggi. Beberapa karakteristik *optimizer* DL ini disediakan seperti dibawah ini dan parameter yang digunakan untuk menyesuaikan fungsi *optimizer* DL dirangkum pada tabel 2.

1. SGD: *Stochastic Gradient Descent* (SGD) merupakan salah satu *optimizer deep learning* yang paling sederhana. Algoritme ini didasarkan pada algoritme *Gradient Descent*. Kecepatan pembelajaran statis untuk semua parameter yang dibutuhkan dalam durasi seluruh pelatihan dan memiliki kemampuan konvergensi yang cepat (Ruder, 2016).
2. Adam: *Adaptive moment estimation Method* (Adam) mengevaluasi kecepatan pembelajaran adaptif dari momen pertama dan kedua dari *gradient* untuk berbagai parameter (Kingma and Ba, 2015).
3. Nadam: *Nesterov-accelerated Adaptive Moment Estimation* (Nadam) merupakan pengembangan dari algoritme Adam dengan menggabungkannya dengan *Nesterov momentum gradient descent* (Dozat, 2016).
4. RMSprop: *Root Mean Square Propagation*, dapat mempercepat *gradient descent* seperti momentum. RMSprop merupakan algoritme *optimizer gradient descent* yang paling populer untuk *deep learning* (Hinton et al., 2012).

Tabel 2. Hyperparameter optimizer deep learning

Optimizer	Parameters
SGD	Learning rate = 0.001, weight decay = 0.0005, momentum = 0.9, nesterov = False
Adam	Learning rate = 0.001, beta1 = 0.9, beta2 = 0.999, epsilon = $1 \times 10^{-8}$ , amsgrad = False
Nadam	Learning rate = 0.001, epsilon = $1 \times 10^{-8}$ , beta1 = 0.9, beta2 = 0.999
RMSprop	Learning rate = 0.001, rho = 0.9, epsilon = 1e-08, decay = 0.0

### Fungsi aktivasi

Fungsi aktivasi yang masih banyak digunakan dalam *deep learning* antara lain Sigmoid, Tanh, ReLU, Leaky ReLU, PReLU. Untuk melatih semua model DL menggunakan fungsi aktivasi ReLU karena efisien secara komputasi (Krizhevsky dan Hinton, 2012). Selain itu fungsi aktivasi Mish yang didasarkan pada penelitian Misra (2019) juga diusulkan dalam penelitian ini. *Input* Mish ditunjukkan variabel ( $k$ ) pada persamaan (2) dan fungsi aktivasi ReLU pada persamaan (3).

$$Mish(k) = k * \tanh(\text{softplus}(k)) \quad (2)$$

$$f(x) = \max(0, x) \quad (3)$$

### Loss Function

*Loss function* merupakan fungsi untuk mengukur seberapa baik performa yang dihasilkan model dengan membandingkan keluaran model. Fungsi koreksi yang paling umum digunakan adalah *loss function*, seperti *Mean Absolute Error* (MAE), *Mean Square Error* (MSE) dan *cross-entropy* dkk., (2019). Pada penelitian ini menggunakan *cross-entropy* seperti yang ditunjukkan pada persamaan (4). Terdiri dari *learning rate* ( $\eta$ ) dan ( $\epsilon$ ) yang menunjukkan keluaran neuron dimana  $L$  menunjukkan keluaran *loss function*.

$$L = -\frac{1}{n} [\eta l \epsilon + (1 - \eta) \ln(1 - \epsilon)] \quad (4)$$

### 2.5. Evaluasi kinerja model

Pada tahap ini dilakukan perhitungan performa dari algoritme klasifikasi CNN. Untuk mengukur kinerja klasifikasi biner menggunakan empat ukuran yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Performa yang dihitung menggunakan parameter ini adalah *accuracy*, *recall*, *precision* dan F1-score, dari klasifikasi biner dihitung menggunakan rumus standar. Karena kasus pada penelitian ini adalah klasifikasi *multiclass* yaitu kelas *hearts* (1), kelas *oblong* (2), kelas *oval* (3), kelas *round* (4), dan kelas *square* (5).

Untuk mengukur kinerja klasifikasi *multiclass*, menghitung performa *recall*, *precision*, F1-score dan *average* akurasi menggunakan rumus (Sokolova and Lapalme, 2009) masing-masing pada persamaan (5), (6), (7) dan (8).

$$Recall = \frac{\sum_{i=1}^l TP_i}{(TP_i + FN_i)} \quad (5)$$

$$Precision = \frac{\sum_{i=1}^l TP_i}{(TP_i + FP_i)} \quad (6)$$

$$F1 - score = \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

$$Average Accuracy = \frac{\sum_{i=1}^l \frac{TP_i + TN_i}{TP_i + FN_i + FP_i + TN_i}}{l} \quad (8)$$

### 2.6. Implementasi

Penelitian ini mengembangkan sistem klasifikasi bentuk wajah dengan teknik *deep learning* yang dijalankan pada sebuah perangkat keras komputer *cloud* dari Google Collaboratory (Google, 2020) dan spesifikasi perangkat keras komputer secara detail ditunjukkan pada tabel 3.

Tabel 3. Spesifikasi perangkat keras komputer untuk klasifikasi *face shape*

Komponen	Spesifikasi
CPU	Intel® Core™ i3-40050 @1.70GHz
RAM (DDR4)	8 GB
Solid State Drive	256 gigabytes
GPU model Name	Nvidia K80 12 GB

Penelitian ini menggunakan empat model CNN yang populer yang diusulkan diantaranya ResNet50, Inception-v3, Inception-ResNet-v2 dan Xception, dengan mempertimbangkan parameter yang lebih efisien, termasuk *optimizer*, *batch size*, *learning rate* dan *loss function*. Dalam eksperimen ini *optimizer* menggunakan RMSprop, SGD, Adam dan Nadam dengan *learning rate* 0.001. Nilai-nilai ini didasarkan pada penelitian Krizhevsky dan Hinton (2012) dan Ruder (2016). Untuk meningkatkan kecepatan proses pelatihan menerapkan *dropout* 0,5 dan *batch size* 32. Selain itu fungsi aktivasi ReLU dan Mish yang merupakan fungsi aktivasi yang paling efektif. *Loss function* adalah *cross entropy*, dan fungsi aktivasi *SoftMax* dengan mengambil bobot dan mengubahnya menjadi probabilitas untuk klasifikasi bentuk wajah. Iterasi sebanyak 50 *epoch* dan *output* terdiri dari lima kelas.

### 3. HASIL PEMBAHASAN DAN DISKUSI

Tabel 4-7 menunjukkan performa model CNN yang dikombinasikan menggunakan fungsi aktivasi dan empat *optimizer* DL. Tabel 4 menggambarkan efektivitas kinerja klasifikasi dari Xception dengan ReLU dibandingkan dengan Xception menggunakan Mish. Hasil analisis dari data pengujian menunjukkan bahwa Xception yang dikombinasikan dengan *optimizer* Nadam dan Mish lebih optimal dari pada Xception yang dikombinasikan dengan ReLU dan SGD dengan perbandingan yang signifikan. Hasil detail dari data pengujian dapat dilihat pada tabel 4.

Tabel 4. Efektifitas kinerja Xception dengan ReLU dan Mish dikombinasikan dengan empat *optimizer*

<i>Optimizer methods</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Waktu (menit)</i>
Adam & ReLU	89.78	90.20	90.00	89.80	56.2
Adam & Mish	90.89	90.80	90.80	90.40	59.0
Nadam & ReLU	91.11	91.40	91.00	91.00	54.0
Nadam & Mish	91.33	91.80	91.20	91.80	64.6
SGD & ReLU	83.11	83.60	83.00	82.80	59.7
SGD & Mish	84.00	83.60	83.20	83.20	49.0
RMSprop & ReLU	88.44	88.60	88.20	88.40	67.0
RMSprop & Mish	90.00	89.80	89.60	89.80	64.2

Tabel 5. Efektifitas kinerja Inception-ResNet-v2 dengan ReLU dan Mish dikombinasikan dengan empat *optimizer*

<i>Optimizer methods</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Waktu (menit)</i>
Adam & ReLU	88.67	88.80	88.60	88.40	49.8
Adam & Mish	89.00	89.20	89.00	89.20	42.4
Nadam & ReLU	87.33	87.40	87.40	87.20	70.3
Nadam & Mish	92.00	92.00	92.00	92.00	65.0
SGD & ReLU	85.56	86.20	85.60	85.40	56.8
SGD & Mish	88.00	87.60	87.20	87.60	44.3
RMSprop & ReLU	88.22	88.60	88.20	88.20	59.9
RMSprop & Mish	89.00	89.00	89.40	89.20	59.3

Tabel 6. Efektifitas Kinerja ResNet50 dengan ReLU dan Mish dikombinasikan dengan empat *optimizer*

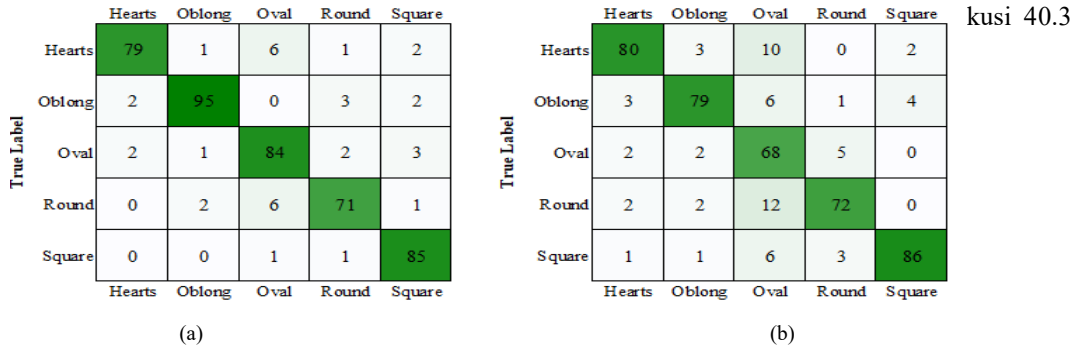
<i>Optimizer methods</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Waktu (menit)</i>
Adam & ReLU	85.11	85.20	85.20	85.20	55.5
Adam & Mish	80.00	81.00	80.40	80.60	43.9
Nadam & ReLU	83.33	83.80	83.60	83.80	55.6
Nadam & Mish	89.80	89.60	89.40	89.00	43.0
SGD & ReLU	87.56	87.40	87.40	87.40	51.8
SGD & Mish	88.00	88.20	88.20	87.80	44.2
RMSprop & ReLU	62.00	61.20	61.60	61.40	78.3
RMSprop & Mish	80.67	80.00	80.60	80.00	67.2

Tabel 7. Efektifitas InceptionV3 dengan ReLU dan Mish dikombinasikan dengan empat *optimizer*

<i>Optimizer methods</i>	<i>Accuracy</i>	<i>Precision</i>	<i>Recall</i>	<i>F1 Score</i>	<i>Waktu (menit)</i>
Adam & ReLU	85.78	85.60	85.80	85.80	39.9
Adam & Mish	88.00	88.00	88.60	88.60	47.9
Nadam & ReLU	87.56	87.20	87.60	87.40	47.5
Nadam & Mish	89.00	89.00	88.80	86.00	43.2
SGD & ReLU	86.00	86.20	86.40	86.00	46.6
SGD & Mish	88.89	89.00	88.60	88.80	40.3
RMSprop & ReLU	46.00	48.00	47.20	46.00	50.0
RMSprop & Mish	82.00	82.40	82.00	82.00	69.6

Inception-ResNetV2 yang dikombinasikan dengan fungsi aktivasi ReLU dan SGD memberikan efektivitas yang terendah dibandingkan Inception-ResNet V2 yang dikombinasikan dengan fungsi aktivasi Mish dan *optimizer* Nadam dengan perbandingan yang signifikan. Hasil detail dari data pengujian dapat dilihat pada tabel 5. Performa Inception-ResNet v2 menggunakan fungsi aktivasi Mish dan ReLU ditunjukkan pada gambar 4.

Gambar 4 menampilkan keefektifan pelatihan model CNN dengan menggunakan dataset pengujian. Hasil *confusion matriks* untuk Inception-ResNet V2 menggunakan kombinasi Nadam dan Mish ditampilkan pada gambar 4a. InceptionResNetV2 mengklasifikasikan bentuk *Hearts* 79 gambar, bentuk *oblong* 95 gambar, bentuk *oval* 84 gambar, bentuk *rounds* 71 gambar dan bentuk *square* 85 gambar. Gambar 4b menunjukkan hasil Inception-ResNetV2 menggunakan kombinasi SGD dan ReLU model ini mengklasifikasikan bentuk *Hearts* 80 gambar, bentuk *oblong* 79 gambar, bentuk *oval* 68 gambar, bentuk *rounds* 72 gambar dan bentuk *square* 86 gambar.



Gambar 4. Hasil Inception dengan menggunakan confusion matrix: (a) Hasil Inception menggunakan Nadam dan Mish;

Tabel 6 menggambarkan efektivitas klasifikasi dari ResNet50 dengan ReLU dibandingkan dengan ResNet50 yang menggunakan Mish. Hasil analisis dari data pengujian menunjukkan bahwa model ResNet50 yang dikombinasikan *optimizer* Nadam dan Mish lebih optimal daripada ResNet50 yang dikombinasikan dengan RMSprop dan ReLU dengan perbandingan yang signifikan. Selain itu, ResNet50 yang dikombinasikan dengan Nadam dan Mish memiliki waktu paling sedikit dibandingkan dengan ResNet50 yang dikombinasikan RMSprop dan ReLU.

InceptionV3 yang dikombinasikan dengan ReLU dan RMSprop menunjukkan kinerja terendah daripada InceptionV3 yang dikombinasikan dengan SGD dan Mish dengan perbandingan yang sangat signifikan. Hasil detail dari data pengujian dapat dilihat pada tabel 7.

Tabel 8. Perbandingan efektivitas kinerja dan penggunaan waktu untuk model klasifikasi menggunakan fungsi aktivasi Mish dan *optimizer* Nadam

CNN model	Accuracy	Precision	Recall	F1 Score	Waktu (menit)
Xception	91.33	91.80	91.20	91.80	64.6
Inception-ResNetV2	92.00	92.00	92.00	92.00	65.0
ResNet50	89.80	89.60	89.40	89.00	43.0
InceptionV3	89.00	89.00	88.80	86.00	43.2

Hasil pengoperasian model klasifikasi menggunakan beberapa pendekatan optimasi dibandingkan pada tabel 8.

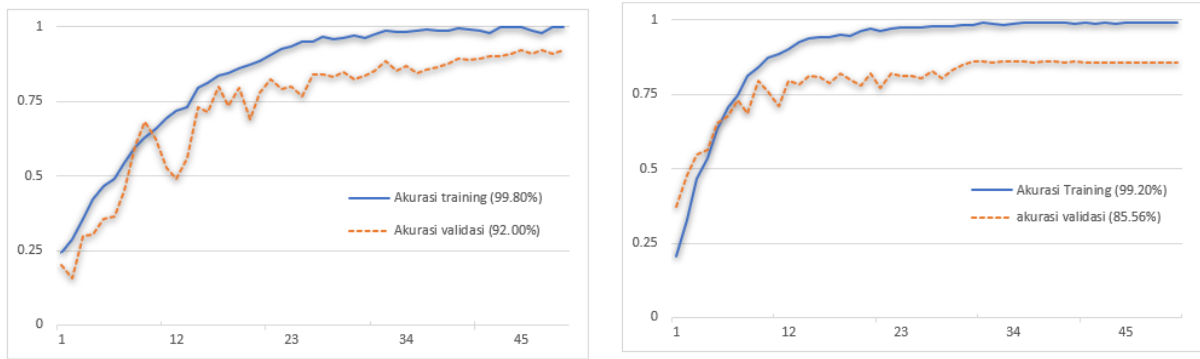
Inception-ResNetV2 yang dikombinasikan dengan Mish dan Nadam merupakan model yang paling akurat, dengan akurasi sebesar 92.00%. Xception dengan Mish dan Nadam memperoleh akurasi sebesar 91.33%, memberikan akurasi terbaik kedua. ResNet50 yang digabungkan dengan Nadam dan Mish memberikan akurasi tertinggi ketiga yaitu 89.80%. ResNet50 dengan Nadam dan Mish dengan akurasi 89.80 %. Waktu minimum diperoleh dari InceptionV3 yang digabungkan dengan Adam dan ReLU, yang memiliki penggunaan waktu eksekusi 39.9 menit saat membandingkan penggunaan waktu pelatihan model klasifikasi. Durasi terpendek berikutnya InceptionV3 yang dikombinasikan dengan

Inception-ResNetV2 dikombinasikan dengan Adam dan Mish memiliki penggunaan waktu terendah ketiga yaitu 42.4 menit. Pada penelitian ini, akurasi dengan fungsi aktivasi Mish lebih tinggi dari pada akurasi dengan fungsi aktivasi ReLU. Mish memiliki batas yang lebih rendah, tetapi tidak ada batas yang lebih tinggi, fitur *seamless* dan *non-monotronic* juga berpengaruh pada produktivitas. Analisis akurasi dan validasi ditunjukkan pada gambar 5 dan 6.

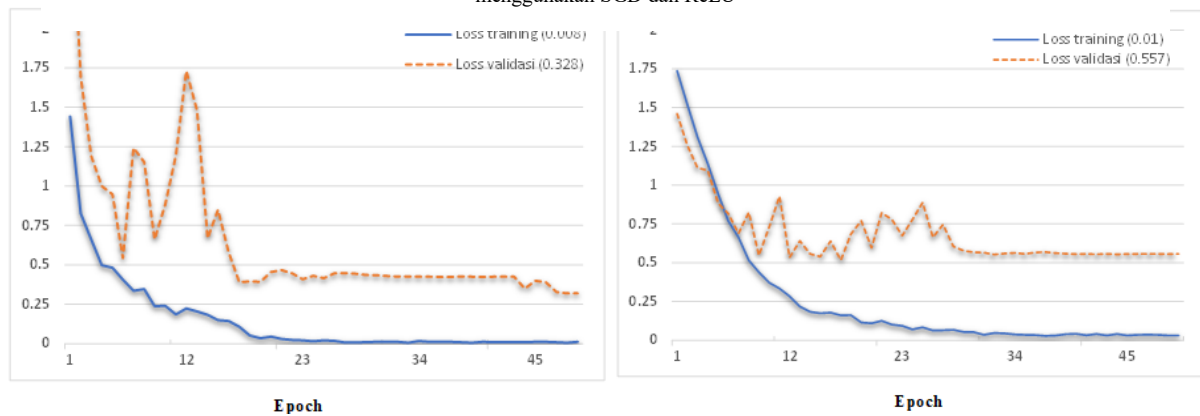
Gambar 5 menunjukkan hasil perbandingan untuk akurasi pelatihan dan akurasi validasi antara Inception-ResNetV2 menggunakan Nadam dan Mish dan Inception-ResNetV2 menggunakan SGD dan ReLU, meningkatkan akurasi menjadi 99.80% untuk pelatihan dan 92.00% untuk validasi.

Gambar 5a menunjukkan bahwa Inception-ResNetV2 dapat meningkatkan presisi menjadi 92.00% dan recall menjadi 92.00%. Pada Gambar 6 Inception-ResNetV2 menggunakan Nadam dan Mish menunjukkan penurunan *loss* 0.008% untuk pelatihan dan 0.382% untuk validasi. Gambar 7a menunjukkan ROC 0.99 dengan model Inception-ResNetV2 yang dikombinasikan Nadam dan Mish dan ROC 0.98 untuk model Inception-ResNetV2 yang dikombinasikan SGD dan ReLU Seperti yang ditunjukkan pada gambar 7b.

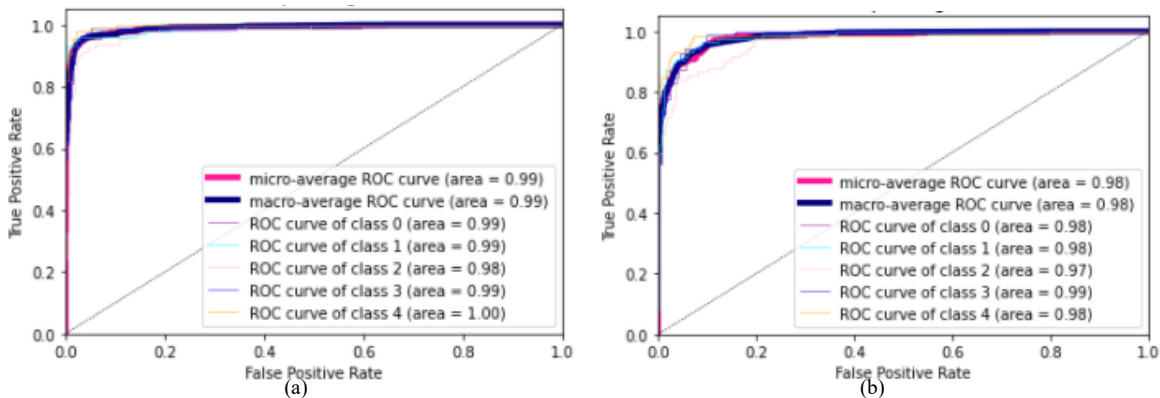
Tabel 9 menampilkan keefektifan pengujian dengan dataset pelatihan, estimasi salah dari kelas *hearts* sebesar 9.09% atau 8 gambar, dari kelas *oblong* sebesar 9.20% atau 8 gambar, dari kelas *oval* sebesar 33.33% atau 34 gambar, dari kelas *round* sebesar 11.11% atau 9 gambar dan dari kelas *square* sebesar 6.52% atau 6 gambar. Untuk hasil klasifikasi model InceptionResNetV2 yang dikombinasikan dengan SGD dan ReLU, estimasi salah dari kelas *hearts* sebesar 4.82% atau 4 gambar, dari kelas *oblong* sebesar 5.05% atau 5 gambar, dari kelas *oval* sebesar 13.4% atau 13 gambar, dari kelas *round* sebesar 8.97% atau 7 gambar dan dari kelas *square* sebesar 8.60% atau 8 gambar.



Gambar 5. History iterasi untuk akurasi validasi dan akurasi training: (a) Akurasi training dan akurasi validasi InceptionResNetV2 menggunakan Nadam dan Mish; (b) Akurasi training dan akurasi validasi InceptionResNetV2 menggunakan SGD dan ReLU



Gambar 6. Histori iterasi untuk loss validasi dan loss training: (a) Loss training dan loss validasi InceptionResNetV2 menggunakan Nadam dan Mish; (b) Loss training dan loss validasi InceptionResNetV2 menggunakan SGD dan ReLU



Gambar 7. Hasil InceptionResNetV2 diperoleh dengan menggunakan kurva ROC dan AUC: (a) InceptionResNetV2 yang dikombinasikan dengan Nadam dan Mish; (b) InceptionResNetV2 yang dikombinasikan dengan SGD dan ReLU

Hasilnya mengungkapkan bahwa penggunaan berbagai metode pengoptimal *deep learning* seperti *transfer learning*, *teknik dropout*, pemilihan *optimizer deep learning*, dan penggunaan fungsi aktivasi Mish dapat meningkatkan kinerja sistem klasifikasi bentuk wajah. Ini juga menunjukkan bahwa pendekatan yang kami usulkan dapat mengungguli metode lain dalam literatur.

Tabel 9. Keefektifan pengujian model CNN menggunakan Inception-ResNetV2 dengan kombinasi Nadam dan Mish

Fungsi aktivasi	Optimizer	Kelas	Jumlah gambar	TP	FP	Efektifitas (%)
ReLU	SGD	Hearts	88	80	8	90.91%
		Oblong	87	79	8	90.80%
		Oval	102	68	34	66.67%
		Round	81	72	9	88.89%
		Square	92	86	6	93.48%
Mish	Nadam	Hearts	83	79	4	95.18%
		Oblong	99	94	5	94.95%
		Oval	97	84	13	86.60%
		Round	78	71	7	91.03%
		Square	93	85	8	91.40%

#### 4. KESIMPULAN

Penelitian ini bertujuan untuk menerapkan model *deep learning* untuk klasifikasi bentuk wajah. Pendekatan yang diusulkan membandingkan arsitektur *deep learning* termasuk Inception-ResNetV2, Xception, InceptionV3, dan ResNet50. Akurasi klasifikasi dapat ditingkatkan dengan penerapan berbagai pengoptimal *deep learning* dan penerapan fungsi aktivasi Mish.

Berdasarkan uji coba yang dilakukan peneliti, algoritma CNN arsitektur Inception-ResNetV2 yang dikombinasikan dengan optimizer Nadam dan fungsi aktivasi Mish memperoleh performa terbaik dalam klasifikasi bentuk wajah. Performa yang didapatkan adalah *accuracy* sebesar 92.0%, *precision* sebesar 92.0%, *recall* sebesar 92.0%, dan *F1-score* sebesar 92.0% dengan penggunaan waktu 65 menit. Menunjukkan bahwa kombinasi model CNN dan algoritma pengoptimal adalah cara paling cocok untuk melakukan klasifikasi bentuk wajah.

#### DAFTAR PUSTAKA

- ALZHRANI, T., AL-NUAIMY, W., & AL-BANDER, B., 2019. Hybrid Feature Learning And Engineering Based Approach For Face Shape Classification, In: 2019 International Conference On Intelligent Systems And Advanced Computing Sciences (ISACS), 2019. Doi:10.1109/ISACS48493.2019.9068910.
- BOUGUILA, J., dan KHOCHTALI, H., 2020. Facial Plastic Surgery And Face Recognition Algorithms: Interaction And Challenges. A Scoping Review And Future Directions, *Journal Of Stomatology, Oral And Maxillofacial Surgery*, 121(6), Pp.696–703. <https://doi.org/10.1016/j.jormas.2020.06.007>.
- CHOLLET, F., 2017. Xception: Deep Learning With Depthwise Separable Convolutions. In: 2017 IEEE Conference On Computer Vision And Pattern Recognition (CVPR), Jul 2017. Doi:10.1109/CVPR.2017.195.
- DOZAT, T., 2016. Incorporating Nesterov Momentum Into Adam. *ICLR Workshop*, (1), Pp.2013–2016.
- FAN, L., ZHANG, T., ZHAO, X., WANG, HAO., & ZHENG, M., 2019. Deep topology network: A framework based on feedback adjustment learning rate for image classification. *Advanced Engineering Informatics*, 42(May), p. 100935. <https://doi.org/10.1016/j.aei.2019.100935>.
- GOOGLE, 2021. Google Colabtraory. [online] Tersedia di: <https://colab.research.google.com/drive/> (Diakses 17 April 2021).
- GRGIC, M., and DELAC, K., 2020. *Face Recognition Homepage*. [Online] Tersedia di: <https://www.Face-ec.Org/Databases/> (Diakses 15 April 2021).
- He, K., ZHANG, X., REN, S., & SUN, JIAN., 2016. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 770–778. Doi:10.1109/CVPR.2016.90.
- HE, Y. L., ZHANG, X.L., AO, W., & HUANG, J.Z., 2018. Determining the optimal temperature parameter for Softmax function in reinforcement learning. *Applied Soft Computing Journal*, 70, pp. 80–85. <https://doi.org/10.1016/j.asoc.2018.05.012>.
- HINTON, G. E., SRIVASTAVA, N., KRIZHEVSKY, A., SUTSKEVER, I., & SALAKHUTDINOV, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv: 1207.0580v1*, pp. 1–18.
- KINGMA, D. P., dan Ba, J. L., 2015. Adam: A method for stochastic optimization. 3rd International Conference on Learning Representations. ICLR 2015 - Conference Track Proceedings, pp. 1–15.
- KRIZHEVSKY, A., dan HINTON, G., 2009. Learning multiple layers of features from tiny images.(2009). *Cs.Toronto.Edu*, pp. 1–58.
- KRIZHEVSKY, A., dan HINTON, G. E., 2012. ImageNet Classification with Deep Convolutional Neural Networks, pp. 1–9.
- LIU, Y., HUANG, M.L., HUANG, W., & LIANG, J., 2017. A physiognomy based method for facial feature extraction and recognition. *Journal of Visual Languages and Computing*, 43, pp. 103–109. <https://doi.org/10.1016/j.jvlc.2017.09.00>.
- MISRA, D., 2019. Mish: A self regularized non-monotonic neural activation function. *Arxiv:1908.08681*.
- PASUPA, K., SUNHEM, W., & LOO, C. K., 2019. A hybrid approach to building face shape classifier for hairstyle recommender system. *Expert Systems with Applications*, 120, pp. 14–32. <https://doi.org/10.1016/j.eswa.2018.11.011>.
- PRAKASH, R.M., THENMOEZHI, N., & GAYATHRI, M., 2019. Face Recognition with Convolutional Neural Network and Transfer Learning. 2019 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 861–864. Doi:10.1109/ICSSIT46314.2019.8987899.
- RUDER, S., 2016. An overview of gradient descent Optimization, pp. 1–14. *Arxiv:1609.04747v2*.
- SARAKON, P., CHAROENPONG, T., &



- CHAROENSIRIWATH, S., 2014. Face Shape Classification From 3D Human Data By Using SVM. *Bmeicon 2014 - 7th Biomedical Engineering International Conference*,  
Doi:10.1109/Bmeicon.2014.7017382.
- SOKOLOVA, M., dan LAPALME, G., 2009. A systematic analysis of performance measures for classification tasks. *Information Processing and Management*, 45(4), pp. 427–437.  
Http://Dx.Doi.Org/10.1016/J.Ipm.2009.03.002.
- SRIVASTAVA, N., HINTON, G., KRIZHEVSKY, A., SUTSKEVER, I., SLAKHUTDINOV, R., 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, pp. 1929–1958.
- SUNHEM, W., dan PASUPA, K., 2016. An Approach To Face Shape Classification For Hairstyle Recommendation. *Proceedings Of The 8th International Conference On Advanced Computational Intelligence, ICACI 2016*, Pp.390–394. Doi: 10.1109/ICACI.2016.7449857.
- SZEGEDY, C., VANHOUCKE, V., LOFFE, S., & SHLENS, J., 2016. Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem, pp. 2818–2826.  
Doi: 10.1109/CVPR.2016.308.
- SZEGEDY, C., LOFFE, S., & VANHOUCKE, V., 2017. Inception-v4, inception-ResNet and the impact of residual connections on learning. 31st AAAI Conference on Artificial Intelligence, AAAI 2017, pp. 4278–4284. Arxiv:1602.07261.
- TIO, A. E. D. C., 2019. Face shape classification using Inception v3.
- ZAFAR, A., dan POPA, T., 2016. Face and eye-ware classification using geometric features for a data-driven eye-ware recommendation system, *Proceedings - Graphics Interface*, pp. 183–188.

*Halaman ini sengaja di kosongkan*