# **BAB IV PENGUJIAN DAN ANALISIS**

Bab ini membicarakan tentang tahap-tahap dalam proses pengenalan aksara Jawa dan hasil dari pengujian pengenalan aksara Jawa dengan menggunakan metode *Transfer Learning*.

## Skenario Simulasi

Skenario simulasi ini dilaksanakan dengan mengimplementasikan model VGG, Inception, dan Xception, masing-masing dengan parameter yang berbeda. Untuk memperjelas arsitektur model transfer learning yang digunakan, berikut adalah gambaran umum mengenai masing-masing model.

A group of colorful rectangular objects

Description automatically generated with medium confidence

KASI FIGURE x.x

Ketiga model, yaitu VGG, Inception, dan Xception, memiliki kesamaan dalam tiga lapisan terakhirnya, yang mencakup proses *pooling*, pengaturan ulang klasifikasi (tuned classifier), dan lapisan *softmax*. Perbedaan utama antara model-model tersebut terletak pada bagian *Feature Extractor* (Ekstraktor Fitur), yang sesuai dengan arsitektur yang telah dijabarkan pada bab sebelumnya.

## Parameter Simulasi

Penulis telah merinci sejumlah parameter yang akan diuji dalam penelitian ini, seperti yang tertera di bawah ini:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **Augmentasi** | **Freeze** | **Learning Rate** | **Layer** | **Optimizer** |
| VGG | Yes, No | Full, ½, None | 0.001, 0.0001 | 1, 2 | Adam, SGD |
| Inception | Yes, No | Full, ½, None | 0.001, 0.0001 | 1, 2 | Adam, SGD |
| Xception | Yes, No | Full, ½, None | 0.001, 0.0001 | 1, 2 | Adam, SGD |

Berdasarkan tabel x, seluruh kombinasi yang ada telah diuji, menghasilkan total 144 skenario yang berbeda. Dalam proses pelatihan (*training*) masing-masing model, penulis juga telah mengimplementasikan *early stopping callback* yang berguna untuk mengehentikan pelatihan model (*training*) jika variabel yang dimonitor tidak mengalami peningkatan performa. Dalam kasus ini variabel tersebut adalah *validation loss*. Hal ini berguna untuk meminimalkan pemborosan sumber daya komputasi yang tidak perlu.

## Matriks Kinerja

Hasil pengujian setiap skenario mencakup informasi tentang waktu pelatihan dan akurasi yang dihasilkan. Untuk menentukan model terbaik dari keseluruhan rangkaian pengujian, penentuan dilakukan berdasarkan akurasi pada set data uji (*test set*) tertinggi, serta waktu pelatihan terpendek yang dicapai.

## *Preprocessing* Data

Tahap *preprocessing* pada data akan melalui berbagai proses yang telah disebutkan pada bab 3. Berikut adalah detail rinciannya.

### Augmentasi

Pada keseluruhan data penulis akan melakukan augmentasi untuk menambah jumlah data yang dapat digunakan di penelitian ini. *Config* atau pengaturan yang digunakan untuk melakukan augmentasi adalah rotasi, *scaling*, dan *blur*.

A screen shot of a computer code

Description automatically generated

Rinciannya terdapat pada gambar x, rotasi memiliki limit 15 yang berarti perubahan pada gambar baru akan mengalami rotasi sebanyak -15 – 15 derajat. *Random scale* mempunyai limit sebanyak 0.7, hal ini berarti gambar *source* akan mengalami penurunan kualitas dalam range 0.7 – 1. *Gaussian blur* mempunyai blur limit default sebanyak (3, 7) detail dari hal tersebut dapat dilihat pada sumber referensi *library* yang digunakan. Variabel p mengindikasikan probabilitas *config* tersebut dapat tereksekusi. Jika nilai p=1 hal ini berarti *config* tersebut akan selalu dijalankan.

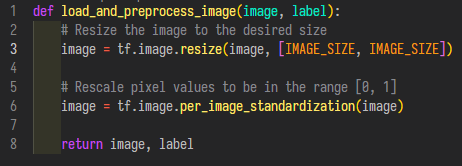
A screen shot of a computer program

Description automatically generated

Gambar x merupakan fungsi yang digunakan untuk melakukan augmentasi. Fungsi tersebut mempunyai 3 parameter yang mengandung informasi terkait *config* augmentasi, *input* *folder*, dan *output folder*. Di dalam fungsi ini, setiap satu gambar *source* akan diaugmentasi sebanyak 5 gambar berbeda. Hasil dari tiap gambar akan berbeda dikarenakan augmentasi dilakukan secara random berdasarkan limit yang telah ditentukan masing – masing *config*. Total sumber data yang telah dibersihkan adalah 3876 gambar dan setelah diaugmentasi, total data menjadi 19128 gambar.

### *Preprocessing Pipeline*

Proses *preprocessing* data selanjutnya akan dilakukan didalam sebuah fungsi *pipeline*. Berikut adalah cuplikan dari *source code*-nya.



Hal yang dilakukan berdasarkan gambar x adalah *resizing* dan standarisasi gambar. Dalam implementasi ini, variabel IMAGE\_SIZE adalah 300, yang berarti setiap gambar akan di-*resize* ke dalam dimensi 300 x 300 pixel.

## *Modelling*

### Pembuatan Model

Proses pembuatan model yang pertama kali adalah melakukan download *pre-trained* model yang dibutuhkan. Dengan *library* TensorFlow, hal ini dapat dilakukan dengan menulis kode saja, berikut adalah rinciannya.

A screenshot of a computer program

Description automatically generated

Gambar x, menunjukkan *source code* yang diperlukan untuk mengunduh 3 model yang sama. Masing – masing model ini nantinya akan dirubah untuk dapat mengimplementasikan *freeze layer* yang sudah disebutkan pada bab 3. Pada model pertama, dilakukan inisialisasi *trainable = False*, ini berarti model tidak akan mengalami pelatihan ulang pada *layer* *feature extractor*. Sebaliknya untuk nilai *True*, model tersebut akan melalui proses pelatihan ulang bagian *layer* *feature extractor*.

Model – model tersebut akan ditambahkan *layer* yang lain (*Pooling, Tuned Classifier,* dan *Softmax*) untuk membuat arsitektur seperti yang telah disebutkan pada skenario simulasi.

A screenshot of a computer

Description automatically generated

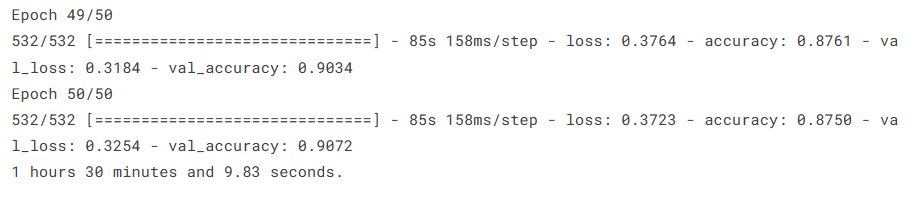
Gambar x adalah salah satu contoh model dengan 2 *tuned classifier*. Model ini mengimplementasikan *full freeze* dengan *feature extractor* model Xception.

### Pelatihan Model

Model – model yang telah dibuat sebelumnya, akan dilatih dengan maksimum *epoch* sebanyak 50. Berikut merupakan cuplikan kode untuk melatih model.

A screenshot of a computer program

Description automatically generated



Pada setiap *epoch*, nilai *accuracy, loss, validation accuracy, dan validation loss* akan disimpan ke dalam sebuah csv yang nantinya akan digunakan untuk menganalisis performa model berdasarkan *epoch* yang telah dilalui. Proses pelatihan juga menggunakan *ModelCheckpoint* untuk menyimpan model dalam setiap *epoch*-nya, untuk menghindari hilangnya model jika terjadi *crash*. Waktu pelatihan model juga akan dihitung dengan fungsi *timer* yang telah dibuat penulis sebagai evaluasi performa model.

## Analisis Hasil Pengujian

Pelatihan setiap model telah terekam dan akan dianalisis dalam bentuk tabel seperti yang terlampir di bawah ini:

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Model** | **Augmentasi** | **Freeze** | **Learning Rate** | **Layer** | **Optimizer** | **Train** | **Validation** | **Test** | **Test Loss** | **Time (min)** | **Epoch** |
| 1 | VGG | Yes | Full | 0.001 | 1 | SGD | 0.21 | 0.55 | 0.51 | 2.63 | 67 | 50 |
| 2 | VGG | Yes | ½ | 0.001 | 1 | SGD | 1.00 | 0.99 | 0.99 | 0.04 | 62 | 30 |
| 3 | VGG | Yes | None | 0.001 | 1 | SGD | 0.99 | 0.99 | 0.99 | 0.03 | 50 | 13 |
| 4 | VGG | Yes | Full | 0.0001 | 1 | SGD | 0.07 | 0.12 | 0.11 | 2.95 | 66 | 50 |
| 5 | VGG | Yes | ½ | 0.0001 | 1 | SGD | 0.97 | 0.98 | 0.98 | 0.07 | 107 | 50 |
| 6 | VGG | Yes | None | 0.0001 | 1 | SGD | 0.96 | 0.98 | 0.98 | 0.06 | 118 | 33 |
| 7 | VGG | Yes | Full | 0.001 | 2 | SGD | 0.23 | 0.53 | 0.49 | 2.38 | 65 | 50 |
| 8 | VGG | Yes | ½ | 0.001 | 2 | SGD | 0.99 | 0.99 | 0.98 | 0.07 | 59 | 28 |
| 9 | VGG | Yes | None | 0.001 | 2 | SGD | 0.99 | 0.99 | 0.98 | 0.09 | 72 | 19 |
| 10 | VGG | Yes | Full | 0.0001 | 2 | SGD | 0.07 | 0.22 | 0.23 | 2.92 | 67 | 50 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |
| 141 | Xception | No | None | 0.001 | 2 | Adam | 0.99 | 1.00 | 0.98 | 0.13 | 15 | 14 |
| 142 | Xception | No | Full | 0.0001 | 2 | Adam | 0.80 | 0.82 | 0.75 | 0.87 | 16 | 50 |
| 143 | Xception | No | ½ | 0.0001 | 2 | Adam | 1.00 | 0.97 | 0.93 | 0.21 | 8 | 14 |
| 144 | Xception | No | None | 0.0001 | 2 | Adam | 1.00 | 1.00 | 0.98 | 0.04 | 14 | 15 |

Untuk memfasilitasi analisis, penulis menggunakan berbagai grafik seperti *line* dan *bar* *chart*. Analisis dilakukan secara komprehensif, meliputi ruang lingkup model serta dampak dari masing-masing parameter yang digunakan.

### Evaluasi Matriks Kinerja

A graph of different colored lines

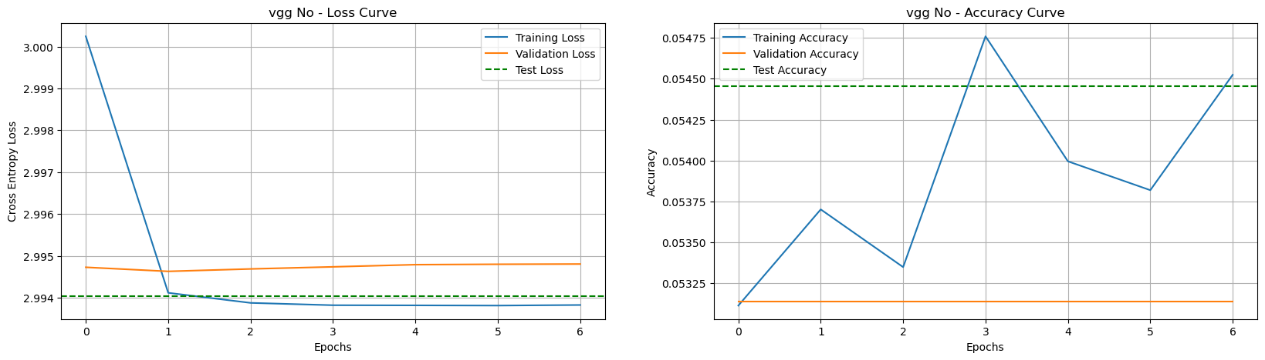
Description automatically generated

Gambar di atas menggambarkan visualisasi dari seluruh skenario yang telah diuji. *Plotting* dilakukan dengan nilai indeks pada tabel sebagai sumbu x (x-axis) dan akurasi uji (*test accuracy*) sebagai sumbu y (y-axis). Hasil *plot* menunjukkan bahwa beberapa model VGG menghasilkan akurasi uji yang stagnan, hanya mencapai sekitar 5% pada beberapa indeks. Penelitian mendalam menunjukkan bahwa model tersebut tidak mengalami peningkatan performa seiring berjalannya setiap epoch. Berdasarkan grafik x terlihat hasil dari pelatihan (*training*) salah satu model VGG tersebut.

A graph of a graph

Description automatically generated with medium confidence

Gambar x



Gambar x

Pada gambar x dan y, terjadi stagnansi pada *validation accuracy* dna *validation loss*.Stagnansi pada pelatihan ini dapat disebabkan oleh ketidakcocokan parameter, dataset, atau arsitektur model yang digunakan dengan konteks penelitian ini (klasifikasi citra huruf tulis tangan aksara Jawa). Hasil skenario juga menunjukkan bahwa penggunaan *optimizer* Adam pada model VGG berpotensi menghasilkan stagnansi dalam pelatihan.

A graph of different colored bars

Description automatically generated

Dari diagram batang (*bar-chart*) di atas, dapat disimpulkan bahwa model VGG memiliki rata-rata akurasi terendah dibandingkan dengan model Inception dan Xception. Namun, perlu diingat bahwa hasil tersebut bisa saja terpengaruh oleh beberapa model yang mengalami stagnansi dalam pelatihan sehingga grafik yang tertera menjadi bias. Grafik di bawah menunjukkan jumlah model dari masing-masing tipe model yang berhasil mencapai akurasi uji di atas 50%.

A graph of different colored bars

Description automatically generated

Gambar x

A graph with blue rectangular bars

Description automatically generated with medium confidence

Gambar x

Berdasarkan grafik tersebut, terlihat bahwa performa model VGG tidak kalah dengan model Inception dan Xception. Namun, jumlah model yang berhasil mencapai kategori akurasi uji di atas 50% lebih banyak pada model Inception, yakni sebanyak 40 model.

A graph of different colored lines

Description automatically generated

Gambar x

A graph of a graph with numbers and a number

Description automatically generated with medium confidence

Gambar x

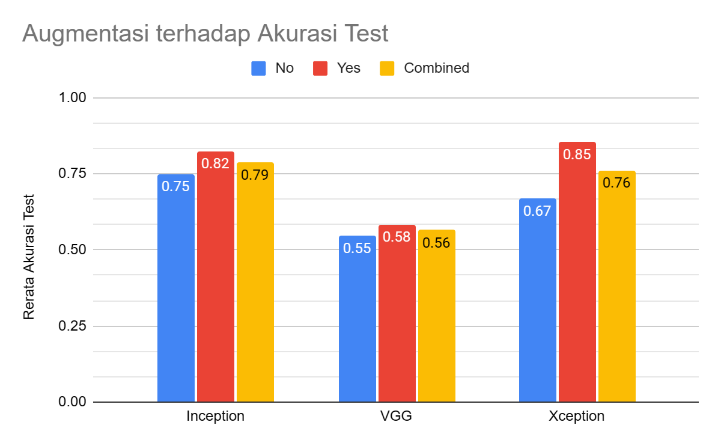
Grafik … (gambar x) di sebelah kiri menunjukkan waktu yang diperlukan oleh seluruh model untuk menyelesaikan pelatihan mereka. Sementara itu, grafik di sebelah kanan menggambarkan rata-rata waktu pelatihan dari masing-masing model. Berdasarkan informasi tersebut, dapat disimpulkan bahwa secara umum, model Inception memiliki waktu pelatihan tercepat dibandingkan dengan model VGG dan Xception. Berikut adalah beberapa contoh hasil uji skenario yang diurutkan berdasarkan kolom Waktu (Time) secara menaik (ascending):

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Model** | **Augmentasi** | **Freeze** | **Learning Rate** | **Layer** | **Optimizer** | **Result Train** | **Validation** | **Test** | **Test Loss** | **Time (min)** | **Epoch** |
| 95 | Inception | No | ½ | 0.0001 | 2 | Adam | 1.00 | 0.98 | 0.95 | 0.17 | 3 | 11 |
| 86 | Inception | No | ½ | 0.001 | 1 | Adam | 0.99 | 0.99 | 0.99 | 0.05 | 3 | 11 |
| 44 | VGG | No | ½ | 0.001 | 2 | Adam | 0.05 | 0.05 | 0.05 | 3.00 | 4 | 8 |
| 139 | Xception | No | Full | 0.001 | 2 | Adam | 0.75 | 0.79 | 0.74 | 0.91 | 5 | 15 |
| 134 | Xception | No | ½ | 0.001 | 1 | Adam | 0.98 | 0.99 | 0.93 | 0.16 | 5 | 9 |
| … | … | … | … | … | … | … | … | … | … | … | … | … |

Hasil uji skenario di atas telah diurutkan berdasarkan kolom Waktu (Time) secara menaik (ascending). Dari data tersebut, terlihat bahwa Inception adalah model yang paling cepat menyelesaikan proses pelatihan hanya dalam 11 epoch dengan total waktu pelatihan selama 3 menit.

### Evaluasi Parameter terhadap Matriks Kinerja

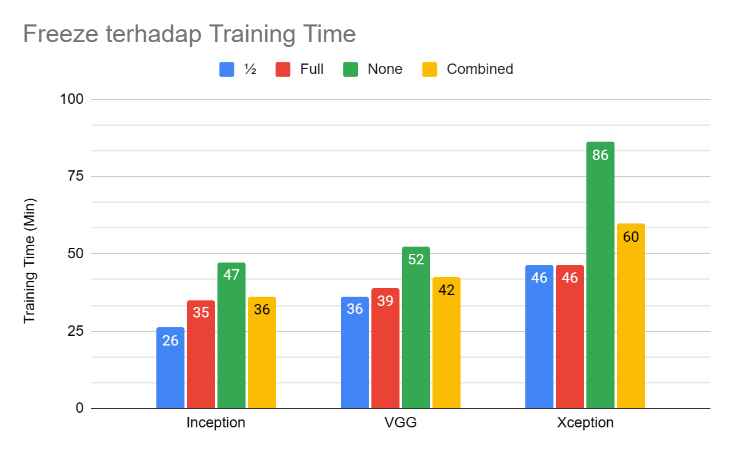
Dalam bagian ini, penulis akan menguraikan parameter-parameter yang digunakan serta dampak masing-masing parameter tersebut pada performa model, terutama dari segi waktu pelatihan dan akurasi set uji (*test set*).

A graph with numbers and a number

Description automatically generated

Parameter augmentasi ini memiliki dampak yang signifikan pada akurasi uji dan waktu pelatihan. Ketika augmentasi diterapkan, dampak terhadap waktu pelatihan menjadi lebih besar. Hal ini dapat dibuktikan dari grafik … terjadi perbedaan signifikan antara nilai parameter “Yes” dan “No”. Dalam kasus ini, Xception mengalami perbedaan waktu pelatihan paling banyak.

A graph of different colored bars

Description automatically generated

Parameter *freeze* dengan nilai "None" memiliki dampak paling signifikan pada akurasi uji untuk model Inception dan Xception. Sebaliknya, pada model VGG, akurasi tertinggi ditemukan pada nilai "Full". Parameter ini juga memengaruhi waktu pelatihan; semakin sedikit lapisan yang tidak di-*freeze* semakin lama waktu pelatihan model tersebut.

A graph of different colored bars

Description automatically generatedA graph of numbers and a number of people

Description automatically generated with medium confidence

Parameter *learning rate* dengan nilai 0.001 menghasilkan akurasi yang lebih tinggi dibandingkan dengan nilai 0.0001. Namun, efek ini hanya berlaku untuk model Inception dan Xception, sementara VGG tidak mengalami perubahan yang signifikan dalam perbedaan nilai parameter ini. Hasil grafik menunjukkan bahwa perbedaan performa akurasi set uji hanya berbanding 1%. Terkait waktu pelatihan model, nilai *learning rate* 0.0001 menghasilkan waktu pelatihan yang lebih lama. Hal ini dikarenakan semakin kecil nilai *learning rate*, semakin lama pula proses konvergensi terjadi dalam tahap pelatihan model.

A graph of different colored bars

Description automatically generatedA graph of numbers and a number of people

Description automatically generated with medium confidence

Parameter *layer* tidak memberikan dampak yang signifikan pada akurasi uji maupun waktu pelatihan model untuk ketiga model tersebut.

A graph of different colored bars

Description automatically generatedA graph of numbers and a number

Description automatically generated with medium confidence

Parameter optimizer ini memiliki dampak signifikan pada akurasi uji dan waktu pelatihan model. Terdapat perbedaan antara model VGG dan dua model lainnya (Inception dan Xception). Model VGG mencapai hasil akurasi uji yang lebih baik dengan nilai parameter SGD dibandingkan dengan Adam. Sedangkan pada kedua model lainnya, Adam menghasilkan akurasi uji yang lebih baik. Selain itu, waktu pelatihan untuk nilai parameter SGD pada umumnya membutuhkan lebih banyak waktu dibandingkan dengan parameter Adam. Model Xception mengalami perbandingan tertinggi untuk waktu pelatihan model. Lama waktu pelatihan dengan nilai SGD hampir mencapai 3x waktu pelatihan dengan nilai parameter Adam.

A colorful squares with text

Description automatically generated with medium confidence

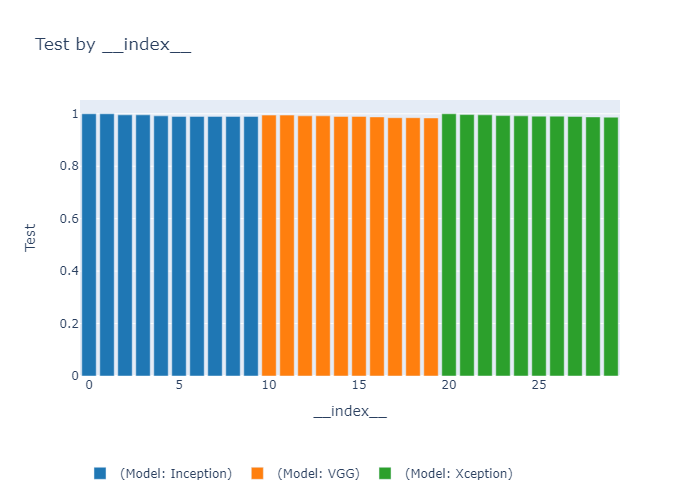
Didukung oleh matriks korelasi (*correlation matrix*), temuan-temuan yang telah disebutkan sebelumnya menguatkan kesimpulan tersebut, membuktikan hubungan antara parameter-parameter tersebut dengan performa model. Sebagai salah satu contoh, variabel Augmentasi\_Yes dan Time (min) mempunyai nilai korelasi positif yang paling tinggi, yaitu sebesar 0,61. Artinya, penggunaan nilai “Yes” pada parameter Augmentasi akan memberikan dampak positif terhadap variabel Time (min). Dengan kata lain, waktu yang dibutuhkan untuk melatih model akan menjadi lebih panjang. Hal ini dapat diulas kembali pada gambar … yang menunjukkan bahwa nilai perbandingan terbesar waktu pelatihan terletak pada parameter Augmentasi.

### Evaluasi Top 10 Models

Pada bagian ini, penulis akan mengulas 10 model terbaik untuk setiap jenis model berdasarkan akurasi uji (*test set*) mereka. Dengan adanya 3 jenis model yang berbeda, total model yang akan dievaluasi adalah 30 model yang berbeda. Tujuan dari analisis ini adalah untuk mendalami parameter-parameter yang paling berpengaruh dalam menciptakan model terbaik. Penulis akan menganalisis parameter-parameter yang dominan digunakan dalam total 30 model terbaik ini. Di bawah ini adalah daftar model yang masuk dalam kategori 10 model terbaik pada masing – masing jenis model.

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Model** | **Augmentasi** | **Freeze** | **Learning Rate** | **Layer** | **Optimizer** | **Result Train** | **Validation** | **Test** | **Test Loss** | **Time (min)** | **Epoch** |
| 65 | Inception | Yes | None | 0.0001 | 1 | Adam | 0.9982 | 0.9961 | 1 | 0.002035 | 38 | 14 |
| 89 | Inception | No | None | 0.0001 | 1 | Adam | 0.9991 | 0.9952 | 1 | 0.004126 | 11 | 19 |
| 62 | Inception | Yes | None | 0.001 | 1 | Adam | 0.9956 | 1 | 0.996 | 0.00795 | 74 | 27 |
| 68 | Inception | Yes | None | 0.001 | 2 | Adam | 0.9949 | 0.9942 | 0.996 | 0.009453 | 52 | 19 |
| 50 | Inception | Yes | None | 0.001 | 1 | SGD | 0.999 | 0.9903 | 0.9921 | 0.02624 | 65 | 23 |
| 56 | Inception | Yes | None | 0.001 | 2 | SGD | 0.9996 | 0.9961 | 0.9901 | 0.03117 | 79 | 31 |
| 74 | Inception | No | None | 0.001 | 1 | SGD | 0.9988 | 0.9952 | 0.9901 | 0.05131 | 28 | 50 |
| 86 | Inception | No | None | 0.001 | 1 | Adam | 0.9882 | 0.9952 | 0.9901 | 0.04759 | 9 | 16 |
| 91 | Inception | No | ½ | 0.001 | 2 | Adam | 0.9962 | 0.9903 | 0.9901 | 0.03443 | 5 | 17 |
| 95 | Inception | No | None | 0.0001 | 2 | Adam | 0.9941 | 1 | 0.9901 | 0.03589 | 8 | 14 |
| 25 | VGG | No | ½ | 0.001 | 1 | SGD | 0.9838 | 0.9903 | 0.995 | 0.1023 | 22 | 44 |
| 40 | VGG | No | ½ | 0.0001 | 1 | Adam | 0.9971 | 1 | 0.995 | 0.01521 | 18 | 35 |
| 1 | VGG | Yes | ½ | 0.001 | 1 | SGD | 0.9952 | 0.9942 | 0.9921 | 0.04196 | 62 | 30 |
| 2 | VGG | Yes | None | 0.001 | 1 | SGD | 0.9871 | 0.9913 | 0.9921 | 0.02506 | 50 | 13 |
| 31 | VGG | No | ½ | 0.001 | 2 | SGD | 0.9471 | 0.9855 | 0.9901 | 0.0664 | 19 | 39 |
| 46 | VGG | No | ½ | 0.0001 | 2 | Adam | 0.9927 | 0.9952 | 0.9901 | 0.02785 | 24 | 50 |
| 11 | VGG | Yes | None | 0.0001 | 2 | SGD | 0.9454 | 0.9826 | 0.9881 | 0.04981 | 185 | 50 |
| 23 | VGG | Yes | None | 0.0001 | 2 | Adam | 0.9931 | 0.9961 | 0.9851 | 0.07386 | 66 | 19 |
| 26 | VGG | No | None | 0.001 | 1 | SGD | 0.9888 | 0.9807 | 0.9851 | 0.1034 | 26 | 28 |
| 7 | VGG | Yes | ½ | 0.001 | 2 | SGD | 0.9918 | 0.9932 | 0.9842 | 0.06673 | 59 | 28 |
| 134 | Xception | No | None | 0.001 | 1 | Adam | 0.9815 | 0.9855 | 1 | 0.01162 | 10 | 9 |
| 113 | Xception | Yes | None | 0.0001 | 1 | Adam | 0.9992 | 0.9981 | 0.997 | 0.01017 | 45 | 10 |
| 104 | Xception | Yes | None | 0.001 | 2 | SGD | 0.9991 | 0.9865 | 0.996 | 0.0247 | 221 | 50 |
| 116 | Xception | Yes | None | 0.001 | 2 | Adam | 0.9912 | 0.9971 | 0.9931 | 0.02588 | 44 | 10 |
| 119 | Xception | Yes | None | 0.0001 | 2 | Adam | 0.999 | 1 | 0.9921 | 0.0427 | 83 | 19 |
| 98 | Xception | Yes | None | 0.001 | 1 | SGD | 0.9994 | 0.9894 | 0.9911 | 0.03065 | 220 | 50 |
| 115 | Xception | Yes | ½ | 0.001 | 2 | Adam | 0.9949 | 0.9981 | 0.9911 | 0.02447 | 37 | 13 |
| 137 | Xception | No | None | 0.0001 | 1 | Adam | 0.9982 | 0.9903 | 0.9901 | 0.02281 | 11 | 11 |
| 110 | Xception | Yes | None | 0.001 | 1 | Adam | 0.9949 | 0.9971 | 0.9881 | 0.0348 | 52 | 12 |
| 109 | Xception | Yes | ½ | 0.001 | 1 | Adam | 0.992 | 0.9971 | 0.9871 | 0.041 | 23 | 8 |

Digambarkan secara *bar-chart*, berikut adalah performa masing – masing 10 model terbaik dalam nilai akurasi set uji (*test set*).



Berikut adalah analisis dampak masing – masing parameter terhadap waktu pelatihan model dan hasil akurasi set uji (*test set*) untuk *top 10 models*.

A graph with red and blue bars

Description automatically generated

Augmentasi: grafik … menggambarkan hasil 10 model terbaik untuk masing-masing jenis model beserta parameter augmentasi (Yes/No). Dapat diperhatikan bahwa 8 dari 10 model terbaik Xception menggunakan augmentasi. Untuk kedua model lainnya (VGG dan Inception), nilai parameter (Yes/No) diambil dengan rasio yang sama.

A graph with red and blue bars

Description automatically generated

Dari grafik … , tidak ada model yang menggunakan nilai "Full" untuk parameter ini. Mayoritas model terbaik menggunakan nilai "½" atau "None". Model terbaik Inception dan Xception rata-rata menggunakan nilai parameter "None", sementara model VGG lebih banyak menggunakan nilai "½." Hal ini menunjukkan bahwa *transfer learning* dengan model yang telah dilatih dari data ImageNet tidak selalu sesuai dengan tugas klasifikasi citra huruf tulisan tangan aksara Jawa. Salah satu alasannya adalah fitur-fitur yang diperoleh dari pelatihan dengan dataset citra ImageNet berbeda dengan citra tulisan tangan aksara Jawa.

A graph with red and blue bars

Description automatically generated

Secara umum, parameter *learning rate* dengan nilai 0.001 dominan digunakan dalam masing – masing 10 model terbaik.

A graph of red and blue bars

Description automatically generated

Berdasarkan gambar … , rata – rata dari 10 model terbaik ini, dihasilkan dengan nilai *layer* 1, terutama untuk model Inception dan Xception. Untuk model VGG, masing – masing nilai *layer* (1 atau 2) memiliki jumlah *count* yang sama. Dengan kata lain, parameter ini tidak terlalu berdampak terhadap performa untuk model VGG.

A graph of different colored bars

Description automatically generated with medium confidence

Sama seperti evulasi parameter keseluruhan model, *top 10 models* Inception dan Xception cenderung lebih cocok dengan nilai parameter *optimizer* Adam. Sedangkan rata – rata model VGG lebih cocok menggunakan *optimizer* SGD. Hal ini dapat diamati dari jumlah hitungan (*count*) masing-masing nilai parameter *optimizer* yang menunjukkan dominasi dibandingkan dengan nilai-nilai lainnya. Ini menunjukkan bahwa pemilihan optimizer memiliki peran penting dalam peningkatan performa model dan dapat berbeda-beda tergantung pada jenis model yang digunakan.

A graph of different colored lines

Description automatically generatedA graph with different colored lines

Description automatically generated

Grafik … dan … merupakan hasil akurasi *training* dan *validation* untuk 10 model terbaik berdasarkan tiap epochnya.

A graph with colorful lines and text

Description automatically generatedA graph with numbers and lines

Description automatically generated

Grafik yang digambarkan di atas mewakili nilai akurasi dan *loss* dari proses pelatihan dan validasi dari masing-masing *top 10 models*. Terlihat jelas bahwa semua model terbaik telah mampu “belajar” secara efektif dari dataset yang digunakan untuk pelatihan. Hal ini dapat disimpulkan dari grafik … , yang secara umum menunjukkan bahwa setiap model telah mengalami konvergensi, dengan peningkatan akurasi yang bertahap dan penurunan *loss* yang landai walaupun terjadi fluktuasi dalam akurasi validasi dan *loss* dalam epoch 1 – 25.

A graph of different colored bars

Description automatically generated

Grafik di atas mewakili jumlah *epoch* yang dilakukan dalam proses pelatihan masing-masing *top 10 models*. Berdasarkan gambar tersebut, hanya lima model yang mencapai 50 *epoch*. Setiap model memiliki rata-rata epoch yang berbeda, menunjukkan kapabilitas masing-masing model dalam proses pelatihan. Sebagai contoh, model Xception sebagian besar hanya membutuhkan kurang dari 20 epoch untuk mencapai kinerja maksimal yang dapat diukur dari pengujian dengan dataset validasi. Namun, kapabilitas masing-masing model dalam pelatihan tidak selalu menentukan kinerja akhir model. Hal ini akan dibahas lebih lanjut pada bagian 4.5, yang akan meninjau kinerja model dengan membuat prediksi dari data di luar sumber pelatihan model (pelatihan, validasi, uji).

A graph with green and orange lines

Description automatically generatedA graph with a line

Description automatically generated

A graph with different colored lines

Description automatically generatedA graph with a line graph

Description automatically generated

Grafik di atas menggambarkan rata-rata akurasi dan *loss* pada pelatihan dan validasi per *epoch*, yang dikelompokkan berdasarkan model. Pengamatan signifikan dari gambar ini terkait dengan model VGG. Model ini menunjukkan konvergensi yang lebih lambat dibandingkan dengan dua model lainnya (Inception dan Xception). Namun, pada akhirnya, model tersebut masih berhasil mencapai kinerja yang setara dengan model Inception dan Xception.

A graph with a line going up

Description automatically generatedA graph with blue and orange lines

Description automatically generated

A graph with a line going up

Description automatically generatedA graph with a line

Description automatically generated

A graph with a line

Description automatically generated

Grafik di atas mewakili dampak masing-masing parameter terhadap akurasi validasi selama *epoch* pelatihannya. Dampak yang dihasilkan dengan nilai yang berbeda dari masing-masing parameter hanya mempengaruhi proses pelatihan model di epoch 1 – 25. Setelah itu, kinerja yang dihasilkan oleh masing-masing model di *epoch-epoch* akhir adalah sama.

Penggunaan augmentasi mempercepat konvergensi di awal proses pelatihan model. Hal ini karena dataset yang digunakan untuk pelatihan di setiap *epoch* lebih besar dibandingkan dengan dataset yang tidak diaugmentasi. Nilai “½” pada *freeze* menunjukkan proses konvergensi yang lebih lambat dibandingkan dengan nilai “None”. Hal ini karena proses pembaruan bobot dalam model dilakukan secara menyeluruh untuk nilai “None” dan hanya setengah untuk nilai “½”. *Learning rate* 0.001 menunjukkan konvergensi yang lebih cepat. Hal ini umum, karena semakin kecil *learning rate*, semakin lama konvergensi terjadi dalam proses pelatihan model. *Layer* dengan nilai 1 menunjukkan konvergensi yang lebih cepat. Hal ini bergantung pada arsitektur model. Karena ada lebih banyak *layer dropout* pada parameter *layer* dengan nilai 2, hal ini mungkin terjadi. *Optimizer* SGD menunjukkan konvergensi yang lebih lambat pada tahap awal pelatihan model. Namun, di sisi lain, Adam mengalami fluktuasi dan penurunan kinerja pada epoch 10 – 20.

Analisis ini memberikan wawasan berharga tentang perilaku berbagai model di bawah berbagai parameter selama proses pelatihan. Penting untuk dicatat bahwa meskipun parameter tertentu dapat mempercepat proses konvergensi, hal tersebut mungkin tidak selalu mengarah ke kinerja yang lebih baik di *epoch* akhir. Oleh karena itu, pertimbangan dan penyetelan yang hati-hati terhadap tiap pameter sangat penting dalam mencapai kinerja model yang optimal.

## Percobaan Model dengan Data Baru

Pada bagian ini, beberapa model yang telah dibuat akan diuji performanya berdasarkan data baru di luar dataset yang digunakan dalam proses pelatihan model. Dengan uji coba ini, penulis berharap untuk memahami sejauh mana model yang telah dilatih dapat menangani data baru yang memiliki karakteristik yang berbeda. Hasilnya diharapkan dapat memberikan wawasan yang berharga dalam mengevaluasi performa model pada aplikasi dunia nyata.

Data baru ini diperoleh melalui tulisan tangan di atas kertas hvs menggunakan bolpoin, kemudian dipindai menggunakan printer, dan dipotong menggunakan perangkat lunak Adobe Photoshop untuk mengambil setiap huruf dari tulisan tangan. Data ini diperoleh dari 30 individu yang memiliki berbagai tingkat pengalaman dalam menulis aksara Jawa. Proses pembersihan data juga dilakukan oleh penulis untuk menghapus data yang tidak sesuai dengan tulisan aksara Jawa.

### Parameter Pengambilan Model

Dalam uji coba ini, *top 10 models* yang telah dianalisis sebelumnya akan digunakan, ditambah dengan *top 3 models* lainnya berdasarkan parameter *freeze*. Pengambilan *top 3 models* ini bertujuan untuk memahami sejauh mana *transfer learning* dapat diimplementasikan dalam kasus ini. Berikut adalah grafik yang menunjukkan hasil akurasi uji pada *top 3 models* tersebut.

A graph of different colored bars

Description automatically generated with medium confidence

Semua model dengan nilai “Full” pada parameter *freeze* tidak terlihat pada *top 10 models*. Namun, kinerja model dengan nilai parameter tersebut tidak terlalu buruk. Untuk model VGG, bisa mencapai akurasi 90%. Hal ini layak ditelusuri lebih lanjut terkait pengujian kinerja model dengan data baru.

### Data

Berikut adalah contoh data sampel baru yang akan digunakan untuk memprediksi citra huruf aksara Jawa. Data ini berbeda dari data yang digunakan selama pelatihan model karena merupakan hasil tulisan tangan di atas kertas. Sedangkan data pelatihan adalah citra tulisan tangan digital. Dengan uji coba ini, penulis bertujuan untuk mengevaluasi kemampuan kinerja model yang telah dilatih. Detail dari data adalah sebagai berikut.

* 1. Data yang telah discan

A sheet of black and white writing

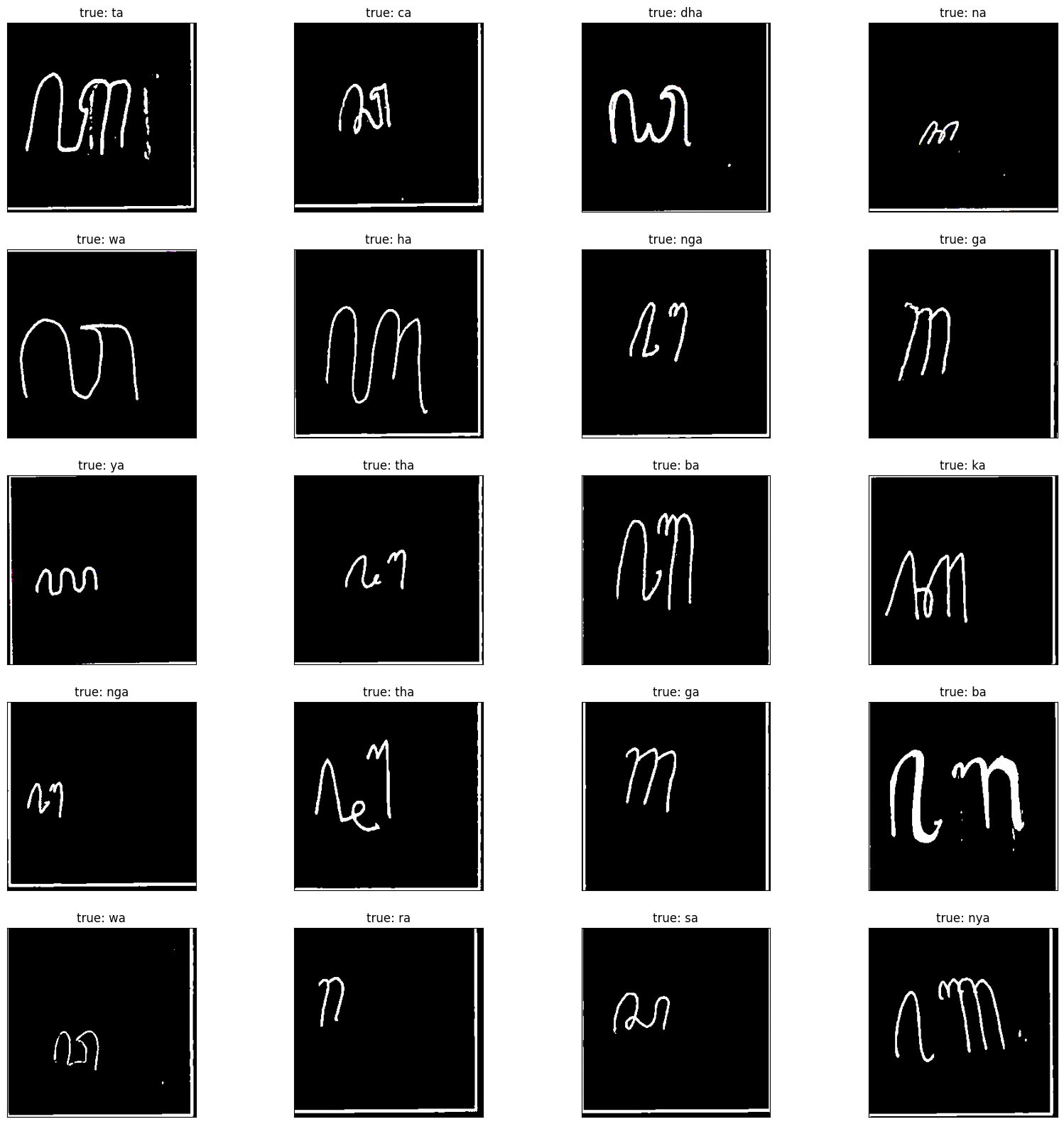
Description automatically generated

* 1. Data yang telah dicrop

A collage of different letters

Description automatically generated

* 1. Data yang sudah melalui tahap *preprocessing*



* 1. Beberapa contoh data yang dibuang (*cleaning*)

A screenshot of a phone

Description automatically generatedA screenshot of a computer screen

Description automatically generatedA screenshot of a signature

Description automatically generated

### Hasil

Pengujian ini dilakukan menggunakan dua pendekatan, yaitu menggunakan *top 10 models* yang telah diidentifikasi sebelumnya dan dengan menggunakan *top 3* *models* berdasarkan nilai parameter *freeze*. Berikut adalah rincian hasilnya dalam format tabel.

#### Menggunakan top 10 models

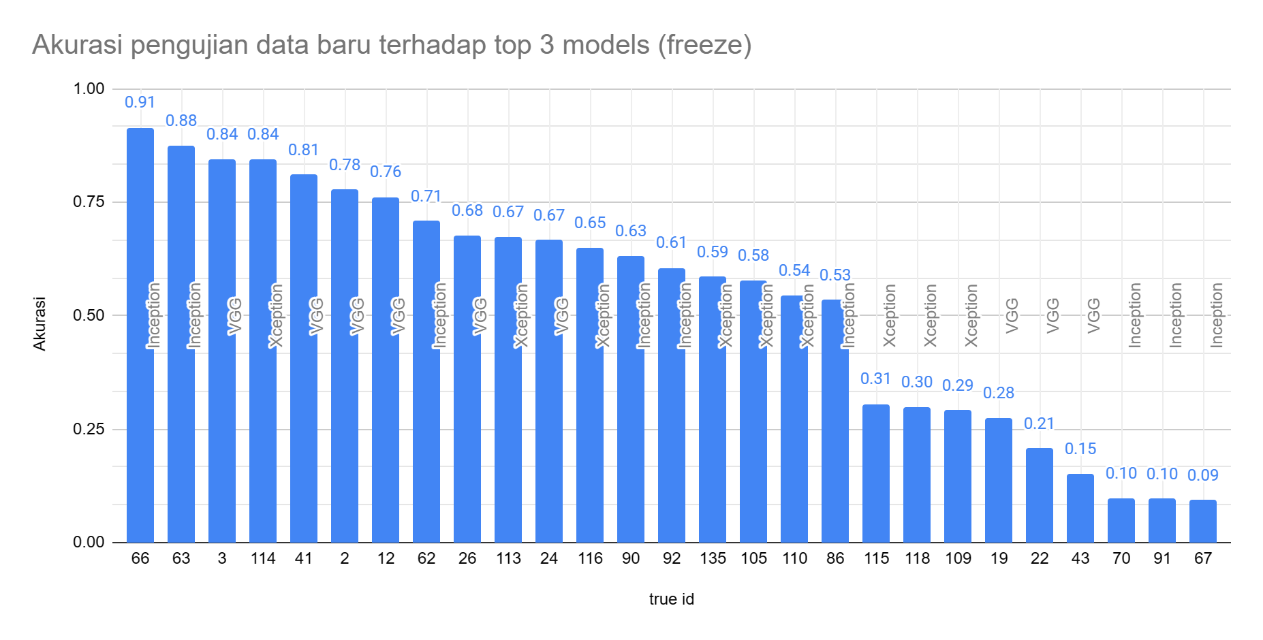
|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Loss** | **Accuracy** | **Model** | **Augmentasi** | **Freeze** | **Learning Rate** | **Layer** | **Optimizer** | **Test** | **Time (min)** | **Epoch** |
| 66 | 0.29 | 0.91 | Inception | Yes | None | 0.0001 | 1 | Adam | 1 | 38 | 14 |
| 63 | 0.56 | 0.88 | Inception | Yes | None | 0.001 | 1 | Adam | 0.996 | 74 | 27 |
| 111 | 0.54 | 0.85 | Xception | Yes | None | 0.001 | 1 | Adam | 0.9881 | 52 | 12 |
| … | … | … | … | … | … | … | … | … | … | … | … |
| 138 | 1.57 | 0.56 | Xception | No | None | 0.0001 | 1 | Adam | 0.9901 | 11 | 11 |
| 110 | 2.19 | 0.54 | Xception | Yes | ½ | 0.001 | 1 | Adam | 0.9871 | 23 | 8 |
| 75 | 1.99 | 0.43 | Inception | No | None | 0.001 | 1 | SGD | 0.9901 | 28 | 50 |

#### Menggunakan top 3 models berdasarkan nilai parameter freeze

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Id** | **Loss** | **Accuracy** | **Model** | **Augmentasi** | **Freeze** | **Learning Rate** | **Layer** | **Optimizer** | **Test** | **Time (min)** | **Epoch** |
| 66 | 0.29 | 0.91 | Inception | Yes | None | 0.0001 | 1 | Adam | 1 | 38 | 14 |
| 63 | 0.56 | 0.88 | Inception | Yes | None | 0.001 | 1 | Adam | 0.996 | 74 | 27 |
| 3 | 0.52 | 0.84 | VGG | Yes | None | 0.001 | 1 | SGD | 0.9921 | 50 | 13 |
| … | … | … | … | … | … | … | … | … | … | … | … |
| 70 | 4.11 | 0.10 | Inception | Yes | Full | 0.0001 | 2 | Adam | 0.7119 | 60 | 50 |
| 91 | 3.76 | 0.10 | Inception | No | Full | 0.001 | 2 | Adam | 0.7079 | 9 | 31 |
| 67 | 3.91 | 0.09 | Inception | Yes | Full | 0.001 | 2 | Adam | 0.7139 | 46 | 39 |

A graph of blue and white bars

Description automatically generated



*Bar-chart* yang digambarkan di atas mewakili kinerja akurasi dari setiap model saat diuji kembali dengan data baru. Dari grafik tersebut, tampak bahwa dua model teratas dari pengujian *top 10 models* dan *top 3 models (freeze)* berasal dari model yang sama yaitu dari indeks 66 dan 63. Perbedaan hasil pengujian model terletak pada posisi ketiga dan seterusnya. Hasil pengujian dengan *top 3 models (freeze)* tidak menunjukkan perbandingan akurasi yang signifikan terhadap *top 10 models*. Bahkan, rata-rata akurasi yang diperoleh dari *top 3 models (freeze)* lebih rendah dibandingkan dengan *top 10 models*. Ini menunjukkan bahwa model-model yang dibuat tanpa nilai “None” (“½”, “Full”) untuk parameter *freeze* tidak menghasilkan kinerja yang sebanding dengan model yang dibuat dengan nilai “None”. Akurasi tertinggi yang dapat dihasilkan oleh model tanpa nilai “None” untuk parameter *freeze* adalah 83%, yang berasal dari model VGG dengan true id 8 dan nilai parameter “½”. Berikut adalah analisis yang dilakukan pada *top 3 models (freeze)* dalam bentuk grafik untuk memperdalam performa masing-masing nilai parameter *freeze*.

A graph of different colored bars

Description automatically generated

Hasil yang diperoleh dari gambar … mencerminkan informasi tentang akurasi maksimum yang dapat dicapai dari prediksi menggunakan data baru, berdasarkan masing-masing nilai parameter *freeze* dan dikelompokkan berdasarkan modelnya. Nilai maksimum yang dicapai dari parameter *full freeze* hanya mencapai 31% pada model Xception. Hal ini memvalidasi informasi yang diperoleh dari evaluasi hasil pengujian sebelumnya, yaitu bahwa fitur yang diperoleh dari *pre-trained model* yang dilatih dengan data ImageNet kurang sesuai dengan fitur yang dibutuhkan dalam kasus penelitian ini (citra tulis tangan aksara Jawa). Dalam konteks parameter *½ freeze*, model masih mampu mengenali 70-80% data baru dengan akurat. Potensi untuk meningkatkan akurasi model dengan konteks tersebut masih ada, mengingat model yang dilatih dalam penelitian ini dibatasi hanya sampai 50 epoch. Untuk parameter *none freeze*, model Inception menunjukkan akurasi maksimum 91%, sedangkan model lainnya hanya mencapai akurasi 84%.

Penulis akan melakukan visualisasi prediksi untuk mengetahui data seperti apa yang membuat model salah prediksi. Berikut adalah beberapa gambar visualisasi dilakukan dengan model id 66 yang merupakan model terbaik dari prediksi dengan data baru.

A collage of images of letters

Description automatically generated

A collage of multiple images of symbols

Description automatically generated

Caption keterangan terkait gambar

Setelah melakukan berbagai analisis, penulis mengamati bahwa data yang diprediksi secara tidak tepat memiliki karakteristik sebagai berikut:

* **Tulisan aksara Jawa yang miring**, yang menyulitkan model untuk mengenali bentuk dan posisi huruf.
* **Tulisan yang tidak ditengah**, yang menyebabkan model untuk mengabaikan sebagian informasi pada gambar.
* **Tulisan yang terlalu kecil**, yang mengurangi resolusi dan kualitas gambar.
* **Tulisan aksara Jawa yang kurang rapi**, yang menimbulkan ambiguitas dan kesalahan dalam pengenalan huruf.

Untuk mengatasi masalah ini, penulis menyarankan untuk meningkatkan *preprocessing pipeline*, misalnya dengan menambahkan algoritma *cropping* yang dapat menyesuaikan ukuran dan posisi huruf secara otomatis. Selain itu, penulis juga menyarankan untuk mengevaluasi dan menyesuaikan *config* augmentasi, agar data *training* lebih mencerminkan variasi dan kondisi data *testing* di dunia nyata. Secara umum, penulis berpendapat bahwa peningkatan performa model dapat dicapai dengan menambahkan data baru yang lebih beragam dan representatif sesuai dengan data di dunia nyata.