**METODE *TRANSFER LEARNING* UNTUK KLASIFIKASI CITRA HURUF TULIS TANGAN AKSARA JAWA**

PROPOSAL TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat Memperoleh Gelar Sarjana Komputer

Program Studi Informatika

A picture containing sketch, symbol, drawing, clipart

Description automatically generated

Diajukan oleh :

Paulus Caesario Dito Putra Hartono

205314159

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS SANATA DHARMA**

**YOGYAKARTA**

**2023**

# **PERSETUJUAN PEMBIMBING**

**PROPOSAL**

**METODE *TRANSFER LEARNING* UNTUK KLASIFIKASI CITRA HURUF TULIS TANGAN AKSARA JAWA**

Disusun oleh :

Paulus Caesario Dito Putra Hartono

205314159

Dosen Pembimbing,

(Ir. Kartono Pinaryanto, S.T., M.Cs.) 16 Juni 2023

# **DAFTAR ISI**

[PERSETUJUAN PEMBIMBING i](#_Toc137327509)

[DAFTAR ISI ii](#_Toc137327510)

[DAFTAR GAMBAR vi](#_Toc137327511)

[DAFTAR TABEL v](#_Toc137327512)

[ABSTRAK vii](#_Toc137327513)

[BAB I PENDAHULUAN 1](#_Toc137327514)

[1.1 Latar Belakang 1](#_Toc137327515)

[1.2 Rumusan Masalah 2](#_Toc137327516)

[1.3 Batasan Masalah 3](#_Toc137327517)

[1.4 Tujuan Penelitian 3](#_Toc137327518)

[1.5 Manfaat Penelitian 3](#_Toc137327519)

[BAB II TINJAUAN PUSTAKA 4](#_Toc137327520)

[2.1 Aksara Jawa 4](#_Toc137327521)

[2.2 Citra Digital 5](#_Toc137327522)

[2.3 Pengolahan Citra Digital 6](#_Toc137327523)

[2.4 Rescale/Resize 6](#_Toc137327524)

[2.5 Augmentasi Data 7](#_Toc137327525)

[2.6 Normalisasi dan Standardisasi 8](#_Toc137327526)

[2.7 Data Splitting 8](#_Toc137327527)

[2.8 Deep Learning 9](#_Toc137327528)

[2.9 Convolutional Neural Network 9](#_Toc137327529)

[2.10 Activation Functions 10](#_Toc137327530)

[2.11 Dropout 10](#_Toc137327531)

[2.12 Pooling 10](#_Toc137327532)

[2.13 Transfer Learning 11](#_Toc137327533)

[2.14 VGG Pre-trained Model 13](#_Toc137327534)

[2.15 Inception Pre-trained Model 14](#_Toc137327535)

[2.16 Xception Pre-trained Model 15](#_Toc137327536)

[2.17 Confusion Matrix 16](#_Toc137327537)

[2.18 Albumentations 16](#_Toc137327538)

[2.19 Tensorflow 16](#_Toc137327539)

[2.20 Review Literatur 17](#_Toc137327540)

[BAB III METODE PENELITIAN 19](#_Toc137327541)

[3.1 Alur Penelitian 19](#_Toc137327542)

[3.2 Data 20](#_Toc137327543)

[3.3 Alat dan Bahan 24](#_Toc137327544)

[3.3.1 Spesifikasi Perangkat Keras 24](#_Toc137327545)

[3.3.2 Libraries 24](#_Toc137327546)

[3.3.3 Preprocessing 25](#_Toc137327547)

[3.3.4 Rescale/Resize 25](#_Toc137327548)

[3.3.5 Normalisasi 26](#_Toc137327549)

[3.3.6 Augmentasi 27](#_Toc137327550)

[3.3.7 Data Splitting 28](#_Toc137327551)

[3.3.8 Modeling 28](#_Toc137327552)

[3.3.9 Download Pre-trained Model 28](#_Toc137327553)

[3.3.10 Apply Transfer Learning 29](#_Toc137327554)

[3.3.11 Fine Tuning 31](#_Toc137327555)

[3.3.12 Train 31](#_Toc137327556)

[3.3.13 Evaluasi 32](#_Toc137327557)

[3.4 Skenario Pengujian 33](#_Toc137327558)

[JADWAL KEGIATAN 34](#_Toc137327559)

[DAFTAR PUSTAKA 36](#_Toc137327560)

# **DAFTAR TABEL**

[Tabel 2.1 Review Literatur 17](#_Toc137328606)

[Tabel 3.1 Skenario Pengujian 33](#_Toc137328612)

# **DAFTAR GAMBAR**

[Gambar 2.1 Aksara Carakan 5](#_Toc137328635)

[Gambar 2.2 Rescale Bilinear Interpolation 6](#_Toc137328636)

[Gambar 2.3 Augmentasi Data 7](#_Toc137328637)

[Gambar 2.4 Ilustrasi max pooling 11](#_Toc137328638)

[Gambar 2.5 Most Common Deep Transfer Learning approaches 12](#_Toc137328639)

[Gambar 2.6 Arsitektur VGG 13](#_Toc137328640)

[Gambar 2.7 Arsitektur Inception 14](#_Toc137328641)

[Gambar 2.8 Arsitektur Xception 15](#_Toc137328642)

[Gambar 3.1 Alur Penelitian 19](#_Toc137328643)

[Gambar 3.2 Alur Preprocessing 19](#_Toc137328644)

[Gambar 3.3 Alur Modeling 20](#_Toc137328645)

[Gambar 3.4 Data sumber pertama 21](#_Toc137328646)

[Gambar 3.5 Data sumber kedua 22](#_Toc137328647)

[Gambar 3.6 Data gabungan 23](#_Toc137328648)

[Gambar 3.7 Contoh rescale 25](#_Toc137328649)

[Gambar 3.8 Contoh standardisasi 26](#_Toc137328650)

[Gambar 3.9 Contoh augmentasi 27](#_Toc137328651)

[Gambar 3.10 Contoh transfer learning 29](#_Toc137328652)

[Gambar 3.11 Contoh pooling 30](#_Toc137328653)

[Gambar 3.12 Contoh dropout 30](#_Toc137328654)

[Gambar 3.13 Contoh confusion matrix 32](#_Toc137328655)

# **ABSTRAK**

Indonesia merupakan negara yang kaya akan budaya, salah satunya adalah aksara Jawa yang merupakan warisan budaya nasional. Namun, saat ini tidak semua orang Jawa dapat membaca aksara Jawa, terutama generasi muda. Oleh karena itu, pengenalan aksara Jawa melalui teknologi dapat menjadi salah satu solusi untuk melestarikan warisan budaya ini. Salah satu teknologi yang dapat digunakan adalah deep learning, yang merupakan salah satu cabang dari machine learning yang dapat digunakan untuk mengenali pola dari data yang kompleks. Convolutional Neural Network (CNN) yang merupakan salah satu jenis network pada deep learning, telah menjadi representative neural networks pada bidang computer vision karena performa yang dihasilkan hampir menyentuh human level accurate. Namun hal tersebut dapat berhasil karena terdapat dataset yang besar. Untuk mengatasi masalah limitasi data, metode *transfer learning* telah berhasil mengatasi masalah keterbatasan data dengan mentransfer pengetahuan yang dipelajari dari satu domain aplikasi ke domain lain yang relevan. Penelitian ini akan mencoba mengimplementasikan dan mencari pre-trained model mana yang terbaik dari VGG, Inception, dan Xception untuk klasifikasi citra huruf tulis tangan aksara Jawa.

Kata Kunci: *Transfer Learning*, Aksara Jawa, *Data Augmentation*, VGG, Inception, Xception

# **BAB I PENDAHULUAN**

## **Latar Belakang**

Indonesia merupakan negara yang memiliki berbagai macam budaya, religi, dan bahasa. Jawa adalah salah satu pulau di Indonesia yang mempunyai populasi tinggi (Abdul Robby et al., 2019). Aksara jawa merupakan salah satu aksara tradisional Indonesia yang berkembang di pulau tersebut. Huruf tersebut masih dapat ditemukan di papan jalanan, tembok, atau peninggalan barang historikal. Aksara jawa juga dianggap sebagai warisan budaya nasional Indonesia. Namun, saat ini masyarakat menghadapi masalah di mana tidak semua orang Jawa dapat membaca aksara jawa, terutama generasi muda (Abdul Robby et al., 2019). Oleh karena itu, pengenalan aksara jawa melalui teknologi dapat menjadi salah satu solusi untuk melestarikan warisan budaya ini.

Salah satu teknologi yang dapat digunakan untuk pengenalan aksara jawa adalah *deep learning*. *Deep learning* merupakan salah satu cabang dari *machine learning* yang dapat digunakan untuk mengenali pola dari data yang kompleks (Lecun et al., 2015). Dalam hal ini, *deep learning* dapat digunakan untuk mengenali pola dari aksara jawa. Berbagai penelitian terkait klasifikasi citra terkhususnya citra tulis tangan telah dilakukan dan menghasilkan akurasi 83% (Ahmed et al., 2022). *Deep learning* telah secara dramatis meningkatkan standar terbaik dalam *speech recognition*, *visual object recognition*, *object detection* dan banyak domain lainnya seperti *drug discovery* dan *genomics* (Lecun et al., 2015).

*Convolutional Neural Network* (CNN) yang merupakan salah satu jenis *network* atau jaringan pada *deep learning*,telah menjadi *representative neural networks* pada bidang *computer vision* karena performa yang dihasilkan dari *network* tersebut hampir menyentuh *human level accurate*  (Chandrarathne et al., 2020; Li et al., 2022). Namun hal tersebut dapat berhasil karena terdapat dataset yang besar (He et al., 2015).

Untuk mengatasi masalah limitasi data, metode *transfer learning* telah berhasil mengatasi masalah keterbatasan data dengan mentransfer pengetahuan yang dipelajari dari satu domain aplikasi ke domain lain yang relevan (Yosinski et al., 2014). Dalam praktiknya, cara umum dari *deep transfer learning* adalah menggunakan CNN yang telah dilatih sebelumnya sebagai model sumber, yang dilatih dengan data dalam jumlah besar seperti ImageNet (Deng et al., 2010). Beberapa model sumber tersebut antara lain adalah Xception, Inception, dan VGG.

Penelitian yang serupa pada bidang citra tulis tangan juga sudah dilakukan oleh berbagai peneliti lainnya. Pada penelitian terdahulu metode *transfer learning* telahmenghasilkan akurasi 98% pada citra aksara jawa dan 91% pada citra aksara sunda (Kesaulya et al., 2022; Khalifa et al., 2022). Dari penemuan tersebut, peneliti akan mencoba mengimplementasikan dan mencari *pre-trained* model mana yang terbaik dari Xception, Inception, dan VGG untuk klasifikasi citra huruf tulis tangan aksara jawa.

## **Rumusan Masalah**

1. Bagaimana cara melakukan implementasi *transfer learning* pada kasus citra huruf tulis tangan aksara jawa?
2. Berapa akurasi yang diperoleh dari metode *transfer learning*?
3. Dari ketiga *pre-trained* model yang peneliti ambil, manakah yang terbaik?

## **Batasan Masalah**

1. Penelitian ini difokuskan untuk klasifikasi pada citra huruf tulis tangan aksara jawa
2. Metode yang digunakan adalah *transfer learning* menggunakan 3 *pre-trained* model yaitu Xception, Inception, dan VGG
3. Data yang digunakan adalah citra huruf tulis tangan aksara jawa tanpa pasangan atau aksara carakan (nglegena)
4. Penelitian ini berfokus untuk mencari *pre-trained* model terbaik dari ketiga model yang diambil peneliti

## **Tujuan Penelitian**

1. Mengklasifikasikan citra huruf tulis tangan aksara jawa (carakan) dengan jumlah total 20 jenis huruf menggunakan metode *transfer learning*
2. Mengevaluasi ketiga *pre-trained* model yang digunakan oleh peneliti

## **Manfaat Penelitian**

1. Mengetahui performa *pre-trained* model terbaik yang diambil oleh peneliti, untuk kasus citra huruf aksara Jawa
2. Membantu orang awam untuk mengenali aksara Jawa
3. Dapat digunakan peneliti lain untuk pengembangan penelitian lebih lanjut

# **BAB II TINJAUAN PUSTAKA**

Bab ini menjelaskan tentang definisi dan teori-teori yang digunakan sebagai landasan penelitian yang berasal dari hasil publikasi dan penelitian dan/atau buku yang relevan.

## **Aksara Jawa**

Aksara Jawa, yang juga dikenal sebagai Hanacaraka dan Dentawyanjana, adalah salah satu aksara tradisional di Indonesia yang berkembang di Pulau Jawa. Aksara ini terutama digunakan untuk menulis bahasa Jawa, tetapi juga digunakan untuk menulis beberapa bahasa daerah lainnya seperti bahasa Sunda, Madura, Sasak dan Melayu serta bahasa historis seperti Sanskerta dan Kawi. Aksara Jawa berasal dari aksara Brahmi India melalui aksara Kawi dan berkerabat dekat dengan aksara Bali. Aksara ini aktif digunakan dalam sastra dan tulisan sehari-hari oleh masyarakat Jawa dari pertengahan abad ke-15 hingga pertengahan abad ke-20 sebelum perlahan-lahan digantikan oleh huruf Latin. Aksara ini masih diajarkan di DI Yogyakarta, Jawa Tengah, Jawa Timur, Cirebon dan Indramayu sebagai bagian dari muatan lokal, tetapi penggunaannya terbatas dalam kehidupan sehari-hari. Aksara Jawa merupakan sistem tulisan *abugida* yang terdiri dari sekitar 20 aksara dasar (Poerwadarminta, 1939; Everson, 2008).

A picture containing text, font, screenshot, number

Description automatically generated

Gambar 2.1 Aksara Carakan

sumber: https://www.rukita.co/stories/aksara-jawa-lengkap/

## **Citra Digital**

Citra digital terdiri dari sejumlah elemen terbatas, masing-masing memiliki lokasi dan nilai tertentu. Elemen-elemen ini disebut elemen citra, pel, dan piksel. Sebuah citra digital dapat didefinisikan sebagai fungsi dua dimensi, f(x,y), di mana x dan y adalah koordinat spasial (bidang), dan amplitudo f pada setiap pasangan koordinat (x,y) disebut intensitas atau tingkat abu-abu citra di titik tersebut. Bidang pengolahan citra digital mengacu pada pengolahan citra digital dengan menggunakan komputer digital (Gonzalez & Woods, 2018)

## **Pengolahan Citra Digital**

Pemrosesan citra adalah kumpulan teknik komputasi untuk menganalisis, meningkatkan, mengompres, dan merekonstruksi citra. Komponen utamanya meliputi impor, di mana sebuah citra ditangkap melalui pemindaian atau fotografi digital; analisis dan manipulasi citra yang dilakukan menggunakan berbagai aplikasi perangkat lunak khusus; serta output (misalnya, ke printer atau monitor). Pemrosesan citra memiliki berbagai aplikasi yang luas di berbagai bidang, termasuk astronomi, kedokteran, robotik industri, dan pemantauan jarak jauh oleh satelit (Gonzalez & Woods, 2018).

## **Rescale/Resize**

Proses *rescale/resize* pada citra merupakan tahap pra-pemrosesan yang kritis dalam visi komputer. Secara prinsip, model *deep learning* dapat dilatih lebih cepat pada citra berukuran kecil. Citra input yang lebih besar memerlukan jaringan saraf untuk belajar dari empat kali lipat jumlah piksel, yang pada akhirnya meningkatkan waktu pelatihan untuk arsitektur tersebut (Saponara & Elhanashi, 2022).

A picture containing diagram, line

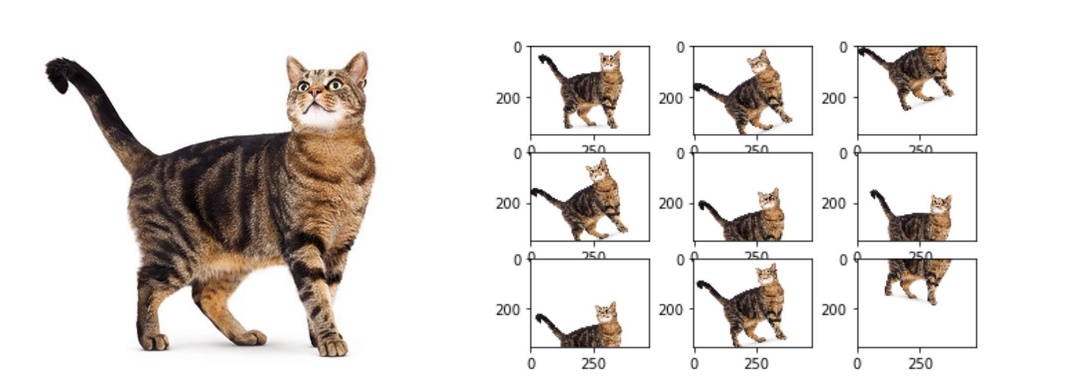
Description automatically generated

Gambar 2.2 Rescale Bilinear Interpolation

sumber: https://theailearner.com/2018/12/29/image-processing-bilinear-interpolation/

## **Augmentasi Data**

Tujuan dari augmentasi data adalah menambahkan titik data baru ke ruang input dengan memodifikasi citra pelatihan sambil mempertahankan informasi semantik dan label target. Oleh karena itu, augmentasi data digunakan untuk mengurangi *overfitting*. Beberapa penelitian mengkonfirmasi pentingnya augmentasi data baik dalam pelatihan maupun pengujian, dan menunjukkan bahwa hal tersebut dapat menghasilkan peningkatan kinerja yang lebih besar daripada mengumpulkan citra sebagai dataset baru (Perez et al., 2018).



Gambar 2.3 Augmentasi Data

sumber: https://analyticsindiamag.com/image-data-augmentation-impacts-performance-of-image-classification-with-codes/

## **Normalisasi dan Standardisasi**

Normalisasi dan standarisasi citra adalah proses yang digunakan dalam pemrosesan citra untuk memastikan perbandingan yang optimal antara metode akuisisi data dan instansi tekstur. Tujuan utama dari normalisasi dan standarisasi citra adalah mengubah piksel citra sehingga mereka dapat dibandingkan secara konsisten dalam berbagai situasi (Gonzalez & Woods, 2018).

(2.1)

(2.2)

## **Data Splitting**

*Data splitting* atau pemisahan data melibatkan pembagian dataset menjadi tiga bagian: set pelatihan, set validasi, dan set pengujian. Set pelatihan digunakan untuk membangun model, sementara set validasi dan pengujian berfungsi sebagai set penahan. Tidak ada proporsi yang pasti untuk pembagian dataset ini, namun aturan umum sebelumnya adalah 70% untuk pelatihan dan 15% untuk validasi dan pengujian. Akan tetapi, dalam era *big data*, proporsi seperti 95% untuk pelatihan dan 2,5%/2,5% untuk validasi/pengujian dapat dipertimbangkan. Dengan menggunakan ketiga subset ini, diharapkan model yang dibangun dapat memprediksi dengan baik contoh-contoh yang tidak dilihat oleh algoritma pembelajaran. Set validasi digunakan untuk memilih algoritma pembelajaran dan menentukan nilai *hyperparameter* terbaik, sementara set pengujian digunakan untuk mengevaluasi model sebelum digunakan di lapangan (Burkov, 2019).

## **Deep Learning**

*Deep learning* adalah cabang dari *machine learning* yang menggunakan jaringan saraf tiruan dengan banyak lapisan tersembunyi untuk memodelkan data yang kompleks (Lecun et al., 2015). Teknik ini memungkinkan komputer untuk mempelajari representasi data yang abstrak dan hierarkis secara otomatis dari data mentah, tanpa perlu pengetahuan domain yang spesifik atau rekayasa fitur yang rumit. *Deep learning* telah mencapai hasil yang sangat baik dalam berbagai tugas pengenalan pola, termasuk pengenalan suara, pengenalan gambar, dan pemrosesan bahasa alami.

## **Convolutional Neural Network**

*Convolutional Neural Network* (CNN) adalah jenis *network* khusus yang secara signifikan mengurangi jumlah parameter dalam *deep neural* *network* dengan banyak unit tanpa kehilangan terlalu banyak kualitas model. CNN telah menemukan aplikasi dalam pengolahan gambar dan teks di mana mereka mengalahkan banyak patokan yang sebelumnya ditetapkan (Burkov, 2019). Dalam implementasi *transfer learning*, CNN bertugas sebagai mengekstrak fitur dari citra data pada model tersebut.

## **Activation Functions**

Untuk memungkinkan jaringan saraf mempelajari batas keputusan yang kompleks, diterapkan fungsi aktivasi non-linier pada beberapa lapisannya. Fungsi yang umum digunakan meliputi *tanh, ReLU, softmax*, dan varian dari fungsi-fungsi tersebut. Secara teknis, setiap neuron menerima sinyal masukan yang merupakan jumlah terbobot dari bobot sinaptik dan nilai aktivasi dari neuron yang terhubung (Zaccone & Karim, 2018).

(2.3)

(2.4)

## **Dropout**

*Dropout* adalah teknik yang digunakan untuk mengurangi *overfitting* dalam jaringan dengan banyak lapisan dan/atau neuron. Secara umum, lapisan *dropout* ditempatkan setelah lapisan yang memiliki sejumlah besar neuron yang dapat dilatih (Zaccone & Karim, 2018).

Teknik ini mengubah pendekatan pembelajaran bobot. Alih-alih mempelajari semua bobot jaringan bersama-sama, *dropout* melatih sebagian dari mereka dalam iterasi pelatihan *batch* (Ranjan, 2019).

## **Pooling**

Fungsi *pooling* dalam *deep learning* adalah untuk mengurangi jumlah koefisien peta fitur yang diproses. Ini dicapai melalui *downsampling*, yang mengurangi dimensi spasial dari input. *Pooling* juga membantu menginduksi hirarki filter spasial dengan membuat lapisan konvolusi berturut-turut melihat jendela yang semakin besar dari segi fraksi input asli yang mereka tutupi. *Max pooling* (salah satu jenis *pooling*) cenderung bekerja lebih baik daripada metode *downsampling* lainnya, seperti *average pooling* atau menggunakan *strides* di lapisan konvolusi sebelumnya, karena lebih informatif untuk melihat kehadiran maksimal fitur yang berbeda daripada kehadiran rata-rata tersebut (Chollet, 2021).



Gambar 2.4 Ilustrasi max pooling

sumber: https://paperswithcode.com/method/max-pooling

## **Transfer Learning**

*Transfer learning* adalah peningkatan pembelajaran dalam tugas baru melalui transfer pengetahuan dari tugas terkait yang sudah dipelajari (Torrey & Shavlik, 2009). *Transfer learning* terdiri dari mengambil jaringan yang sudah dibangun dan membuat perubahan yang sesuai pada parameter dari berbagai lapisan sehingga dapat menyesuaikan dengan dataset lain (Zaccone & Karim, 2018).

**A picture containing text, screenshot, font, diagram

Description automatically generated**

Gambar 2.5 Most Common Deep Transfer Learning approaches

(Iman et al., 2023)

Pendekatan populer pertama adalah menyesuaikan model yang telah dilatih pada data target, hal ini merupakan metode *Deep Transfer Learning* (DTL) yang paling umum digunakan karena kemudahannya. Metode ini dapat meningkatkan pelatihan pada data target dengan mengurangi biaya pelatihan dan mengatasi kebutuhan dataset target yang luas, tetapi masih rentan terhadap pelupakan yang mematikan. Metode ini telah diterapkan pada dataset gambar dan tabular di berbagai bidang. Pendekatan populer kedua adalah membekukan lapisan CNN dalam model yang telah dilatih dan menyesuaikan hanya lapisan terhubung sepenuhnya lateral. Dalam metode ini, lapisan CNN mengekstrak fitur dari dataset yang diberikan, dan *fully connected layer* bertanggung jawab untuk klasifikasi dan akan disesuaikan dengan tugas baru untuk data target (Iman et al., 2022).

## **VGG Pre-trained Model**

Visual Geometric Group (VGG) adalah model arsitektur jaringan saraf tiruan yang sangat dalam dan dikembangkan oleh Visual Geometry Group di Universitas Oxford (Simonyan & Zisserman, 2014). Model ini terdiri dari beberapa lapisan konvolusi yang diikuti oleh lapisan *max-pooling* dan beberapa lapisan sepenuhnya terhubung di bagian akhir. Salah satu keunggulan utama dari model VGG adalah penggunaan filter konvolusi berukuran kecil (3x3) di semua lapisannya, yang memungkinkan jaringan untuk memiliki kedalaman yang lebih besar dengan jumlah parameter yang sama dibandingkan dengan jaringan yang menggunakan filter konvolusi berukuran lebih besar. Model VGG mencapai hasil yang sangat baik pada *benchmark* klasifikasi ILSVRC 2012 dan telah digunakan secara luas sebagai titik awal untuk berbagai tugas pengenalan gambar.

**A picture containing text, screenshot, parallel, number

Description automatically generated**

Gambar 2.6 Arsitektur VGG

(Simonyan & Zisserman, 2014)

## **Inception Pre-trained Model**

Inception adalah model arsitektur jaringan saraf tiruan yang digunakan untuk visi komputer. Model ini dirancang untuk meningkatkan kualitas jaringan dengan cara yang efisien secara komputasi dengan menggunakan konvolusi yang difaktorkan dan regularisasi agresif (Szegedy et al., 2015). Inception-v3 adalah salah satu versi dari model Inception yang mencapai kinerja tinggi pada *benchmark* klasifikasi ILSVRC 2012 dengan biaya komputasi yang relatif rendah dibandingkan dengan arsitektur yang lebih sederhana dan monolitik (Szegedy et al., 2016). Model ini juga menggunakan jumlah parameter yang lebih sedikit dan regularisasi tambahan dengan *classifier auxilary* yang dinormalisasi *batch* dan *label-smoothing* untuk melatih jaringan berkualitas tinggi pada set pelatihan berukuran relatif sedang (Szegedy et al., 2016).

**A picture containing text, screenshot, number, font

Description automatically generated**

Gambar 2.7 Arsitektur Inception

(Szegedy et al., 2016)

## **Xception Pre-trained Model**

Xception adalah arsitektur jaringan saraf konvolusional yang sepenuhnya didasarkan pada lapisan konvolusi terpisah secara mendalam *(depthwise separable convolution layers).* Arsitektur ini diusulkan sebagai perpanjangan dari arsitektur Inception, di mana modul Inception telah digantikan dengan konvolusi terpisah secara mendalam. Arsitektur Xception adalah tumpukan linier lapisan konvolusi terpisah secara mendalam dengan koneksi residual (Chollet, 2016).

**A screenshot of a computer program

Description automatically generated with low confidence**

Gambar 2.8 Arsitektur Xception

(Chollet, 2016)

## **Confusion Matrix**

Confusion Matrix adalah tabel yang merangkum seberapa sukses model klasifikasi dalam memprediksi contoh yang termasuk dalam berbagai kelas. Salah satu sumbu dari *confusion matrix* adalah label yang diprediksi oleh model, dan sumbu lainnya adalah label sebenarnya. Dalam masalah klasifikasi biner, terdapat dua kelas (Burkov, 2019).

## **Albumentations**

Albumentations adalah *library* Python untuk augmentasi citra yang cepat dan fleksibel. *Library* ini secara efisien mengimplementasikan berbagai operasi transformasi gambar yang kaya dan dioptimalkan untuk kinerja, sambil memberikan antarmuka augmentasi gambar yang ringkas namun kuat untuk berbagai tugas visi komputer, termasuk klasifikasi objek, segmentasi, dan deteksi (Buslaev et al., 2020).

## **Tensorflow**

TensorFlow adalah antarmuka untuk mengekspresikan algoritma *machine learning* dan implementasi untuk menjalankan algoritma tersebut. Sistem ini fleksibel dan dapat digunakan untuk mengekspresikan berbagai macam algoritma. TensorFlow telah digunakan untuk penelitian dan penerapan sistem pembelajaran mesin di berbagai bidang. API TensorFlow dan implementasi referensi dirilis sebagai paket sumber terbuka pada November 2015 dan tersedia di [www.tensorflow.org](http://www.tensorflow.org) (Abadi et al., 2016).

## **Review Literatur**

Tabel 2.1 Review Literatur

| **No** | **Judul** | **Penulis (Tahun)** | **Metode** | **Hasil** |
| --- | --- | --- | --- | --- |
| 1 | *Transfer Learning Implementation on Sundanese Script Recognition Using Convolutional Neural Network* | MUHAMMAD KHALIFA U, et al. (2022) | *Transfer learning* dengan *pre-trained* model dari dataset *alphabet*, *devanagari*, *arabic*, dan aksara jawa | Model *transfer learning* terbaik didapat dari *pre-trained* dataset *arabic,* mencapai akurasi 91,86% dan *loss* 0.2814 |
| 2 | Javanese Script Text Image Recognition Using Convolutional Neural Networks | Goldy Najma Adli Kesaulya, et al. (2022) | *Transfer learning* dengan *pre-trained* model ResNeXt dengan melakukan *freezing* di 4 layer pertama dari 10 layer model *pre-trained* | Akurasi testing mencapai 98.19% |
| 3 | Deep Learning for Image Classification on Very Small Datasets Using Transfer Learning | Menying Shu (2019) | *Fine-tuned pre-trained* model VGG16, VGG19, InceptionV3, InceptionResNetV2 | Akurasi testing mencapai 96% dengan menggunakan *fine-tuned* model dari InceptionResNetV2 |
| 4 | A Close Look at Deep Learning with Small Data | L Brigato, et al. (2020) | Review metode yang dapat digunakan untuk menyelesaikan permasalahan *small data*, menggunakan berbagai variasi dari model CNN dan ResNet | *Regularization technique* seperti *data augmentation* dan dropout dapat meningkatkan performa dari *deep learning* |
| 5 | A Comprehensive Study on Deep Image Classification with Small Datasets | Gayani Chandrarathne, et al. (2019) | Review metode *transfer learning* dengan *fine-tuned pre-trained* model VGG-16 pada berbagai macam dataset. | *Fine-tuning* (*re-init few layers*)dan *(whole network)* menunjukkan performa yang signifikan dibandingkan dengan *scratch training.* Hasil akurasi yang diperoleh dari *fine-tuning* adalah 91.4%, 95.52% dan 79.6% secara terurut |
| 6 | Understanding the Mechanisms of Deep Transfer Learning for Medical Images | Hariharan Ravishankar, et al. (2017) | CNN *transfer learning* CaffeNet | *Transferred* dan *fine-tuned* model dapat menungguli performa *state-of-the-art* *feature engineered pipeline* (Haar) dan menghasilkan akurasi 85% |
| 7 | Transfer Learning using CNN for Handwritten Devanagari Character Recognition | Nagender Aneja, et al. (2019) | *Transfer learning* AlexNet, DenseNet, VGG, dan Inception | InceptionV3 menghasilkan akurasi sebesar 99% |
| 8 | A Deep Learning-Based Framework for Automatic Brain Tumors Classification Using Transfer Learning | Arshia Rehman, et al. (2019) | *Transfer learning* menggunakan *pre-trained* model AlexNet, GoogLeNet, dan VGG16 dari dataset ImageNet | *Fine-tuned* VGG16 menghasilkan akurasi tertinggi sebesar 98.69% |
| 9 | A Review of Deep Transfer Learning and Recent Advancements | Mohammadreza Iman, et al. (2023) | Review berbagai metode *deep transfer learning* yang sudah diteliti dalam waktu dekat | (i) *Finetuning*, (ii) *Freezing* CNN *Layers*, dan (iii) *Progressive Learning* adalah teknik yang telah terbukti kemampuan dan efektivitasnya untuk berbagai masalah *machine learning*. |
| 10 | Text recognition on images using pre-trained CNN | Afgani Fajar Rizky, et al. (2023) | *Transfer leraning* dengan *augmentation, freeze-layers,* dan *fine-tuning* telah dilakukan dengan menggunakan model AlexNet, VGG, ResNet, dan DenseNet | Akurasi terbaik dihasilkan dari model VGG dengan 0 *freeze layer* dan *augmentation* |
| 11 | Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images | Srikanth Tammina (2019) | *Transfer learning* menggunakan *augmentation, fine-tuning,* dan *freeze layers* dengan *pre-trained* model VGG | Hasil terbaik diperoleh dari proses *fine-tuning* dan *augmentation* dengan akurasi 95.40% pada validation |

# **BAB III METODE PENELITIAN**

## **Alur Penelitian**

Penelitian akan dilakukan dengan bahasa pemrograman Python dan berbagai *library* yang disebutkan pada bab 3.3.2. Platform yang digunakan untuk mengimplementasikan penelitian adalah Kaggle Notebooks. Untuk mencapai tujuan yang telah dipaparkan pada bab I. Berikut flowchart penelitian yang dirancang untuk menjelaskan langkah - langkah yang akan diambil di penelitian ini, yang akan didetailkan pada sub bab 3.2 dan seterusnya.

A picture containing text, font, screenshot, graphics

Description automatically generated

Gambar 3.1 Alur Penelitian

Secara garis besar, terdapat 4 tahap utama dari penelitian ini. Tahap pre-processing dan modeling akan diperjelas pada diagram flowchart dibawah ini.

A picture containing screenshot, line, text, font

Description automatically generated

Gambar 3.2 Alur Preprocessing

A picture containing text, screenshot, line, rectangle

Description automatically generated

Gambar 3.3 Alur Modeling

Detail dari kedua diagram diatas akan dijelaskan lebih lanjut pada bab 3.3.3 dan seterusnya.

## **Data**

Data yang peneliti gunakan diambil dari kaggle. Kaggle adalah sebuah platform komunitas *data science* yang berisi berbagai sumber dataset, notebook, kompetisi, dan lain sebagainya. Terdapat dua sumber dataset yang peneliti ambil yaitu,

1. https://www.kaggle.com/datasets/phiard/aksara-jawa  
   dengan total data 2634 sebagai sumber pertama
2. <https://www.kaggle.com/datasets/vzrenggamani/hanacaraka>  
   dengan total data 1579 sebagai sumber kedua

Data ini akan digabung menjadi satu sebagai dataset yang akan diuji oleh peneliti. Total data yang akan digunakan di dalam penelitian ini adalah 4213 data.

A picture containing text, diagram, sketch, design

Description automatically generated

Gambar 3.4 Data sumber pertama

A picture containing text, font, black and white, typography

Description automatically generated

Gambar 3.5 Data sumber kedua

A picture containing text, diagram, font, design

Description automatically generated

Gambar 3.6 Data gabungan

## **Alat dan Bahan**

1. **Spesifikasi Perangkat Keras**

Perangkat yang digunakan pada penelitian ini diambil dari platform notebook kaggle. Kaggle menyediakan jupyter notebook yang bisa digunakan untuk menjalankan kode dengan menggunakan GPU, CPU, dan RAM yang cukup untuk mengimplementasikan *deep learning*. Berikut detail spesifikasinya:

* + 1. *20 GB of auto-saved disk space*
    2. *1 Nvidia Telsa P100 GPU*
    3. *2 CPU cores*
    4. *13 Gigabytes of RAM*

Lebih lengkapnya dapat dilihat di dokumentasi kaggle: https://www.kaggle.com/docs/notebooks

1. **Libraries**

*Library* yang akan digunakan pada penelitian ini mencakup visualisasi, *preprocessing*, *modeling*, dan evaluasi. Tensorflow, Albumentations, Numpy, Pandas, Matplotlib, Plotly, Sklearn akan digunakan di dalam penelitian ini.

1. **Preprocessing**

Dari data tersebut akan dilakukan *preprocessing* (lihat **Error! Reference source not found.**) untuk mengoptimalkan pelatihan model yang digunakan nantinya. Pada penelitian ini, dilakukan berbagai skenario penelitian untuk melihat perbandingan hasil dari berbagai proses yang dilakukan. Pada bagian ini, augmentasi data akan dilakukan sesuai dengan skenario pengujian peneliti.

1. **Rescale/Resize**

Dalam tahap ini citra akan melalui tahap *rescale* ke dalam ukuran 300 x 300. Hal ini dilakukan agar data yang akan digunakan untuk proses *training* seragam. Citra yang masuk ke dalam proses *training* juga diharuskan seragam, dalam data sebelumnya, beberapa citra mempunyai dimensi yang tidak sama dengan citra lainnya. Berikut adalah contoh rescale dari citra dimensi 364 x 300 menjadi 300 x 300.

A picture containing diagram, sketch, line, drawing

Description automatically generated

Gambar 3.7 Contoh rescale

1. **Normalisasi**

Di tahap ini, peneliti akan menggunakan teknik standardisasi pada citra. Hal ini dilakukan untuk mengoptimalkan dan menjaga kualitas data yang akan digunakan pada proses *training* nantinya. Berikut adalah contoh standardisasi data yang dilakukan dengan *library* tensorflow.

A picture containing text, sketch, handwriting, design

Description automatically generated

Gambar 3.8 Contoh standardisasi

1. **Augmentasi**

Dalam tahap ini, dilakukan augmentasi data dengan *config* seperti pada penelitian sebelumnya terkait data citra teks (Rizky et al., 2023). Berikut confignya:

* *Rotation: 15 °*
* *Image Scale: 0.9*
* *Blur Effect*

A picture containing line, font, sketch, white

Description automatically generated

Gambar 3.9 Contoh augmentasi

1. **Data Splitting**

Peneliti akan melakukan tiga split menjadi *train, validation*, dan *test set* dari total data yang ada dengan perbandingan 0.9 : 0.05 : 0.05. Total data awal adalah 4242, hal ini akan menghasilkan *data spli*t sebesar 3817, 212, 212 secara terurut dari *train, validation,* dan *test set*.

1. **Modeling**

Tahap *modeling* (lihat **Error! Reference source not found.**) akan dilakukan sesuai skenario pengujian yang tertera. Berikut detail dari tiap prosesnya.

1. **Download Pre-trained Model**

Semua pre-trained model ini akan diperoleh menggunakan *library* keras tensorflow. *Keras Applications* menyediakan berbagai *pre-trained* model yang dapat digunakan untuk penggunaan lebih lanjut seperti *prediction, feature extraction,* dan *fine-tuning*. Dokumentasi lebih lanjut dapat dilihat di <https://keras.io/api/applications/>. Model yang akan diunduh dari *keras application library* adalah VGG, Inception, dan Xception.

1. **Apply Transfer Learning**

Untuk melakukan transfer learning diperlukan kustomisasi pada layer yang diunduh dari *pre-trained* model keras *library*. Dalam beberapa penelitian sebelumnya, *transfer learning* dengan *pre-trained* model VGG dilakukan dengan menambahkan *fully connected layer* dan *output layer* dengan *activation function* softmax (Tammina, 2019).

...

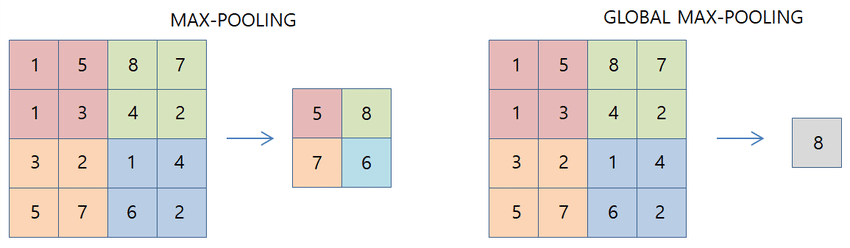
A picture containing text, screenshot, font, line

Description automatically generated

Gambar 3.10 Contoh transfer learning

(Tammina, 2019)

Untuk menjaga kualitas model dan mencegah *overfitting,* digunakan juga *pooling layer* dan *dropout layer* pada lapisan *fully connected layer*. Ilustrasi dari kedua *layer* tersebut adalah sebagai berikut.



Gambar 3.11 Contoh pooling

sumber: <https://www.researchgate.net/figure/The-difference-of-max-pooling-and-global-max-pooling_fig4_338079465>

A picture containing diagram, line, circle, origami

Description automatically generated

Gambar 3.12 Contoh dropout

(Zaccone & Karim, 2018)

1. **Fine Tuning**

Definisi dari *fine-tune* menurut *Oxford Languages* adalah membuat penyesuaian kecil (sesuatu) untuk mencapai yang terbaik atau kinerja yang diinginkan. Dalam konteks *deep transfer learning*, hal ini dapat dicapai dengan melakukan *freezing layer, train fully connected layer*, maupun *train all the network* (Iman et al., 2022). Pelaksanaan *fine-tuning* lebih detailnya dapat dilihat pada tabel skenario pengujian.

1. **Train**

Pada proses ini, dilakukan pelatihan model *transfer learning* dengan data yang sudah dilakukan *preprocessing*. Di dalam *training*, digunakan data *train* dan *validation* untuk mengukur nilai *overfitting* yang terjadi.

1. **Evaluasi**

Evaluasi akan dilakukan dengan metriks akurasi, sebagai tambahan digunakan juga *confusion matrix* untuk melihat kesalahan prediksi yang dilakukan oleh model yang sudah di-*train*. Berikut adalah contoh *confusion matrix* dengan 6 kelas prediksi. Dalam kasus penelitian ini, total kelas adalah 20, sehingga total baris dan kolom masing – masing adalah 20.

A picture containing screenshot, square, rectangle

Description automatically generated

Gambar 3.13 Contoh confusion matrix

(Zaccone & Karim, 2018)

## **Skenario Pengujian**

Dalam tahap penelitian akan dilakukan evaluasi terhadap skenario pengujian dengan mengimplementasikan variabel augmentasi dan *freezing* *layer* dengan tiga model berbeda yaitu VGG, Inception, dan Xception. *Config* dari parameter augmentasi akan disesuaikan dengan penelitian terdahulu yang sudah disinggung pada bab 3.3.6. Skenario secara garis besar adalah seperti tabel berikut. Total pengujiannya adalah 18.

Tabel 3.1 Skenario Pengujian

|  |  |  |
| --- | --- | --- |
| **MODEL** | **AUGMENTASI** | ***FREEZE*** |
| VGG | *Yes No* | *Full ½ None* |
| Inception | *Yes No* | *Full ½ None* |
| Xception | *Yes No* | *Full ½ None* |

# **JADWAL KEGIATAN**

Tahun: 2023

Nama: Paulus Caesario Dito Putra Hartono

NIM: 205314159

Prodi: Informatika

Jenjang: S1

Judul: METODE TRANSFER LEARNING UNTUK KLASIFIKASI CITRA HURUF TULIS TANGAN AKSARA JAWA

| NO. | KEGIATAN PENELITIAN | JADWAL/BULAN | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| APRIL | | | | MEI | | | | JUNI | | | | JULI | | | | AGUSTUS | | | | SEPTEMBER | | | | OKTOBER | | | | NOVEMBER | | | | DESEMBER | | | |
| 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 | 1 | 2 | 3 | 4 |
| 1 | Pengumpulan Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | Penyusunan Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | Seminar Proposal |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | *Preprocessing* Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | Uji Coba tanpa Augmentasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | Augmentasi Data |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | Uji Coba dengan Augmentasi |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | Analisa dan Hasil |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | Evaluasi dan *Debugging* |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | Penyusunan bab 4 dan 5 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

# **DAFTAR PUSTAKA**

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., … Research, G. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*. https://arxiv.org/abs/1603.04467v2

Abdul Robby, G., Tandra, A., Susanto, I., Harefa, J., & Chowanda, A. (2019). Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application. *Procedia Computer Science*, *157*, 499–505. https://doi.org/10.1016/J.PROCS.2019.09.006

Ahmed, R. M., Rashid, T. A., Fattah, P., Alsadoon, A., Bacanin, N., Mirjalili, S., Vimal, S., & Chhabra, A. (2022). Kurdish Handwritten character recognition using deep learning techniques. *Gene Expression Patterns*, *46*, 119278. https://doi.org/10.1016/J.GEP.2022.119278

Burkov, A. (2019). The Hundred-Page Machine Learning Book. Andriy Burkov.

Buslaev, A., Iglovikov, V. I., Khvedchenya, E., Parinov, A., Druzhinin, M., & Kalinin, A. A. (2020). Albumentations: Fast and flexible image augmentations. *Information (Switzerland)*, *11*(2). https://doi.org/10.3390/INFO11020125

Chandrarathne, G., Thanikasalam, K., & Pinidiyaarachchi, A. (2020). A Comprehensive Study on Deep Image Classification with Small Datasets. *Lecture Notes in Electrical Engineering*, *619*, 93–106. https://doi.org/10.1007/978-981-15-1289-6\_9/COVER

Chollet, F. (2016). Xception: Deep Learning with Depthwise Separable Convolutions. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, *2017-January*, 1800–1807. https://doi.org/10.1109/CVPR.2017.195

Chollet, F. (2021). Deep Learning with Python (2nd ed.). Manning Publications Co.

Deng, J., Dong, W., Socher, R., Li, L.-J., Kai Li, & Li Fei-Fei. (2010). *ImageNet: A large-scale hierarchical image database*. 248–255. https://doi.org/10.1109/CVPR.2009.5206848

Everson, M. (2008, March 6). Proposal for encoding the Javanese script in the UCS (N3319R3) [PDF]. ISO/IEC JTC1/SC2/WG2. Unicode.

G., A. R., Tandra, A., Susanto, I., Harefa, J., & Chowanda, A. (2019). Implementation of Optical Character Recognition using Tesseract with the Javanese Script Target in Android Application. Procedia Computer Science, 499-505.

Gonzalez, R. C., & Woods, R. E. (2018). Digital Image Processing (4th ed., Global ed.). Harlow, England: Pearson Education Limited.

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 770–778. https://doi.org/10.1109/CVPR.2016.90

Iman, M., Rasheed, K., & Arabnia, H. R. (2022). A Review of Deep Transfer Learning and Recent Advancements. *Technologies*, *11*(2), 40. https://doi.org/10.3390/technologies11020040

Kesaulya, G. N. A., Fariza, A., & Karlita, T. (2022). Javanese Script Text Image Recognition Using Convolutional Neural Networks. *IES 2022 - 2022 International Electronics Symposium: Energy Development for Climate Change Solution and Clean Energy Transition, Proceeding*, 534–539. https://doi.org/10.1109/IES55876.2022.9888527

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature 2015 521:7553*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Li, Z., Liu, F., Yang, W., Peng, S., & Zhou, J. (2022). A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(12), 6999–7019. https://doi.org/10.1109/TNNLS.2021.3084827

Perez, F., Vasconcelos, C., Avila, S., & Valle, E. (2018). Data Augmentation for Skin Lesion Analysis. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11041 LNCS*, 303–311. https://doi.org/10.1007/978-3-030-01201-4\_33

Poerwadarminta, W.J.S. (1939). Baoesastra Djawa (dalam bahasa Jawa). J.B. Wolters.

Ranjan, C. (2019). Understanding Deep Learning: Application in Rare Event Prediction. Chitta Ranjan.

Rizky, A. F., Yudistira, N., & Santoso, E. (2023). *Text recognition on images using pre-trained CNN*. https://arxiv.org/abs/2302.05105v1

Saponara, S., & Elhanashi, A. (2022). Impact of Image Resizing on Deep Learning Detectors for Training Time and Model Performance. *Lecture Notes in Electrical Engineering*, *866 LNEE*, 10–17. https://doi.org/10.1007/978-3-030-95498-7\_2/COVER

Simonyan, K., & Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*. https://arxiv.org/abs/1409.1556v6

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *07-12-June-2015*, 1–9. https://doi.org/10.1109/CVPR.2015.7298594

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, *2016-December*, 2818–2826. https://doi.org/10.1109/CVPR.2016.308

Tammina, S. (2019). Transfer learning using VGG-16 with Deep Convolutional Neural Network for Classifying Images. *International Journal of Scientific and Research Publications (IJSRP)*, *9*(10), p9420. https://doi.org/10.29322/IJSRP.9.10.2019.P9420

Torrey, L., & Shavlik, J. (2009). Chapter: Transfer Learning. Hershey, Pennsylvania: IGI Global.

U, M. K., Sihabuddin, A., & S.Si., M.Kom., Dr. (n.d.). Transfer learning implementation on Sundanese script recognition using convolutional neural network (Bachelor’s thesis).

Yosinski, J., Clune, J., Bengio, Y., & Lipson, H. (2014). How transferable are features in deep neural networks? *Advances in Neural Information Processing Systems*, *4*(January), 3320–3328. https://arxiv.org/abs/1411.1792v1

Zaccone, G., & Karim, M. R. (2018). Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python (2nd ed.). Birmingham, United Kingdom: Packt Publishing.