

# Pengembangan Aplikasi Klasifikasi Gambar Menggunakan Library Tensorflow yang Menerapkan Algoritma Convolutional Neural Network Studi Kasus: Galeri Foto Kegiatan Ibadah Gereja Shoot Fellowship

Willyam Saputra<sup>1)</sup>, Yulius Denny Prabowo<sup>2)</sup>

Informatika, Fakultas Ilmu Komputer dan Desain, Institut Teknologi dan Bisnis Kalbis  
Jalan Pulomas Selatan Kav. 22, Jakarta 13210

<sup>1)</sup>Email: willyamsaputra@gmail.com

<sup>2)</sup>Email: yulius.prabowo@kalbis.ac.id

**Abstract:** This research is aimed to research and develop a desktop application which will classify images of church activities using the Convolutional Neural Network method to group image data so as to facilitate image search. The dataset used in this research is image data of church activities of SHOOT Fellowship. The software development flow used is an incremental model and the user interface will be created using the Tkinter framework.

**Keywords:** classification, cnn, incremental, pray, sermon, tkinter.

**Abstrak:** Untuk naskah dalam bahasa Indonesia, abstrak dibuat dalam dua bahasa, yaitu Inggris dan Indonesia berisi maksimal 150 kata menggunakan font Times New Roman 10 point Italic. Abstrak berisi maksud dan tujuan penelitian, metode penelitian yang digunakan serta hasil yang dicapai. Kata kunci berupa kata atau bagian kalimat disusun menurut abjad dan dipisahkan dengan tanda koma. Jumlah kata kunci antara 4 dan 6 menggunakan huruf kecil semua.

**Kata kunci:** berdoa, cnn, inkremental, khotbah, klasifikasi, tkinter.

## I. PENDAHULUAN

Program untuk mengklasifikasikan gambar sudah diterapkan untuk banyak hal, seperti klasifikasi emosi atau perasaan manusia, klasifikasi jenis kendaraan, klasifikasi hewan, dan kali ini peneliti tertarik untuk menggunakan klasifikasi gambar untuk mengklasifikasikan gambar kegiatan ibadah, sehingga data gambar dapat dikelompokkan berdasarkan kegiatan yang sedang dilakukan.

Data gambar yang disimpan setelah kegiatan ibadah tidak terorganisir dengan baik, data gambar tidak dikelompokkan berdasarkan kegiatan yang sedang dilakukan. Sehingga ketika

melakukan pencarian gambar yang diinginkan cukup memakan waktu yang lama karna gambar harus dilihat satu-persatu dan tidak dapat dicari berdasarkan judul atau nama gambar karna nama data gambar yang dihasilkan kamera biasanya memiliki nama dengan format tertentu dan itu pun juga tidak menjelaskan kegiatan dalam gambar.

Setelah melakukan observasi dan wawancara pada tim media gereja SHOOT Fellowship ternyata mereka mengalami kesulitan saat mencari data gambar, dan pada saat membutuhkan gambar yang diinginkan biasanya mereka mencari data gambar tersebut satu-persatu, dan judul atau nama gambar pun tidak dapat dijadikan patokan untuk mencari gambar karna judul atau nama gambar masih berbentuk kode format kamera sehingga

pencarian cukup memakan banyak waktu.

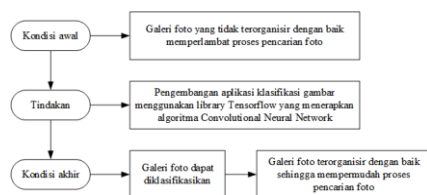
Oleh karena itu mengklasifikasikan gambar kegiatan ibadah dapat membantu mengelompokkan data gambar berdasarkan kegiatan yang sedang dilakukan. Diharapkan nantinya mengelompokkan data gambar ini dapat mempermudah proses pencarian gambar yang diinginkan.

Sebelum melakukan penelitian ini, ada pun penelitian terdahulu yang digunakan sebagai referensi. Penelitian terdahulu yang dijadikan sebagai referensi ini diambil dari jurnal Anwar Hossain dengan judul “Classification of Image using Convolutional Neural Network (CNN)” pada tahun 2019 lalu. Selain judul diatas, ada beberapa jurnal juga yang menjadi referensi bagi peneliti yaitu tulisan dengan judul “An Analysis of Convolutional Neural Networks For Image Classification” pada tahun 2018 lalu, dan beberapa referensi lainnya.

Penelitian ini akan mengklasifikasikan gambar kegiatan ibadah menggunakan metode Convolutional Neural Network untuk mengelompokkan data gambar sehingga mempermudah pencarian gambar. Data yang digunakan didalam penelitian ini menggunakan data yang dibuat sendiri oleh peneliti dan diamati secara langsung oleh peneliti agar hasil klasifikasi gambar yang didapat lebih akurat dan tepat sesuai dengan foto yang terdapat pada gereja SHOOT Fellowship.

## II. METODE PENELITIAN

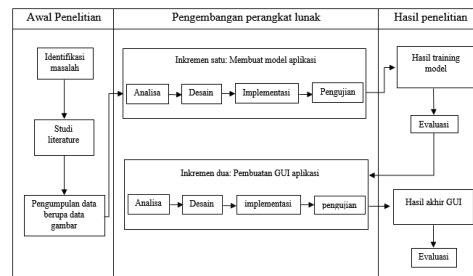
### A. Kerangka Pemikiran



Gambar 1 Kerangka pemikiran

Penyimpanan data gambar yang tidak terorganisir dengan baik membuat pencarian gambar yang diinginkan menjadi sulit dan cukup memakan waktu, dan untuk mengatasi permasalahan tersebut, peneliti akan melakukan pengembangan perangkat lunak yang mampu mengklasifikasi atau mengelompokkan data gambar berdasarkan kegiatan ibadah yang sedang dilakukan, sehingga diharapkan nantinya dapat mempermudah proses pencarian gambar yang diinginkan.

### B. Proses Penelitian



Gambar 2 Proses penelitian

Tahap awal penelitian dimulai dari identifikasi masalah. Masalah yang akan diangkat adalah penyimpanan data gambar yang tidak terorganisir dengan baik membuat pencarian gambar yang diinginkan menjadi sulit dan cukup memakan waktu. Oleh sebab itu, penelitian ini bertujuan untuk pengembangan perangkat lunak yang mampu mengklasifikasi atau mengelompokkan data gambar berdasarkan kegiatan ibadah yang sedang dilakukan.

Di dalam proses penelitian ini, peneliti membutuhkan teori-teori dasar untuk mendukung metode yang digunakan. Oleh sebab itu, peneliti melakukan *studi literature* dimulai dari buku, *e-book*, prosiding serta jurnal-jurnal penelitian terdahulu yang berkaitan dengan metode yang digunakan oleh peneliti. Selain peneliti membutuhkan *studi literature*, peneliti juga membutuhkan data objek penelitian

berupa gambar yang terdapat pada penyimpanan gereja SHOOT Fellowship.

Selanjutnya pada tahap proses penelitian dilakukan pengembangan perangkat lunak. Pada tahap ini, peneliti akan menggunakan metode inkremental. Metode inkremental ini akan menggunakan 2 tahapan yaitu, tahap inkremen yang pertama dilakukan untuk membuat dan melatih model menggunakan dataset. Model akan menghasilkan prediksi kategori gambar.

Tahap inkremen kedua dilakukan untuk pengembangan *graphic user interface* pada model yang telah dibuat. Pengujian dilakukan terhadap seluruh komponen *graphic user interface* agar sesuai dengan fungsinya dan dapat mencapai tujuan yang diharapkan.

### C. Ibadah

Ibadah adalah pelayanan atau pernyataan bakti kepada Tuhan [1, hlm. 58], dalam hal ini dapat diwujudkan dengan melakukan perintah-perintah yang diajarkan maupun mengikuti upacara atau kegiatan kebaktian. Dalam pengajaran kristen protestan, kegiatan kebaktian dilakukan di gereja dan meliputi beberapa kegiatan seperti berdoa, penyembahan, khotbah, baptis, kesaksian, dan lainnya. Dalam kegiatan kebaktian setiap gereja dapat berbeda-beda mengenai tata cara dan pelaksanaannya, tergantung ketentuan gereja dan sinode yang dianut.

Ibadah pada hakikatnya adalah suatu interaksi antara Tuhan dengan manusia, dan interaksi ini seharusnya melahirkan tekad pada manusia untuk mempersembahkan seluruh hidupnya bagi Tuhan [1, hlm. 59-60]. Salah satu cara untuk berinteraksi dengan Tuhan adalah dengan berdoa yaitu berbicara kepada Tuhan dan mendengarkan perintah Tuhan melalui khotbah yang dibagikan.

### D. Artificial Intelligence

*Artificial intelligence* (Kecerdasan Buatan) adalah salah satu bagian ilmu komputer yang mempelajari bagaimana komputer dapat melakukan pekerjaan seperti yang dilakukan manusia bahkan lebih baik dari manusia [2, hlm. 17]. Dasarnya adalah membuat komputer dapat mengambil keputusan dengan pemikiran selayaknya manusia. Dalam pembuatan kecerdasan buatan terdapat dua hal yang menjadi bagian utama yaitu *knowledge base* dan *inference engine*. *Knowledge base* berisi tentang fakta-fakta, teori, pemikiran dan hubungan antara satu dengan yang lainnya dan *inference engine* berisi kemampuan menarik kesimpulan berdasarkan pengalaman.

### E. Machine Learning

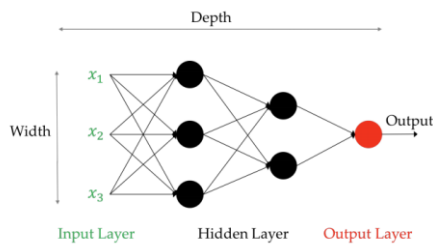
*Machine learning* (Pembelajaran Mesin) adalah salah satu bagian dari Kecerdasan buatan, yang berfokus pada bagaimana komputer dapat membuat keputusan berdasarkan data yang telah dipelajari sebelumnya. Terdapat beberapa metode pembelajaran dalam pembuatan *machine learning* yaitu:

- *Supervised learning* adalah metode pembelajaran yang menggunakan masukan data pembelajaran yang telah diberi label. Setelah itu membuat prediksi dari data yang telah diberi label [3, hlm. 1042].
- *Unsupervised learning* adalah metode pembelajaran yang menggunakan masukan data pembelajaran yang tidak diberi label. Setelah itu mencoba untuk mengelompokan data berdasarkan karakteristik-karakteristik yang ditemui [3, hlm. 1051].

- *Reinforcement learning* adalah metode pembelajaran yang menggunakan data pembelajaran dan tes yang saling dicampur. Untuk mengumpulkan informasi pembelajaran secara aktif dengan berinteraksi ke lingkungan sehingga untuk mendapatkan balasan untuk setiap aksi dari pembelajar [4, hlm. 3].

## F. Artificial Neural Network

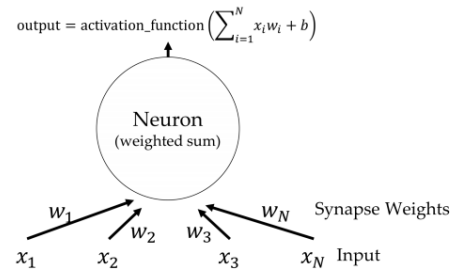
*Artificial neural network* muncul pertama kali ketika manusia ingin menggabungkan antara kemampuan otak manusia yang mampu untuk belajar dan komputer yang memiliki kemampuan untuk memproses banyak data dan menyimpannya. Karena banyak sekali masalah yang tidak mampu dirumuskan oleh algoritma biasa.



Gambar 3 Artificial neural network

### 1. Single Perceptron

Bentuk terkecil sebuah *artificial neural network* adalah *single perceptron* yang hanya terdiri dari sebuah *neuron*. Secara matematis, terdapat fitur vektor  $x$  yang menjadi input bagi *neuron* tersebut. Fitur vektor mempresentasikan suatu data *point, event, atau instance*. *Neuron* akan memproses *input*  $x$  melalui perhitungan jumlah perkalian antara *input* dan *synapse weight*, yang dilewatkan pada fungsi *non-linear*. keluaran yang dihasilkan *neuron* merupakan fungsi aktivasi.

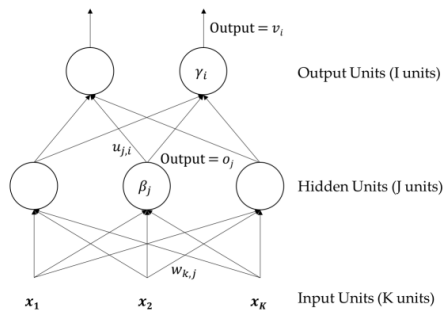


Gambar 4 Single perceptron

### 2. Multi Layer Perceptron

*Multi layer perceptron* memiliki beberapa *layers*, secara umum ada tiga *layers* penting pada *multi layer perceptron* yaitu *input, hidden, output layers*. *Input layer* menerima *input*, kemudian nilai *input* dikirim ke *hidden units*. Pada *hidden units*, *input* diproses dan dilakukan perhitungan hasil fungsi aktivasi untuk tiap – tiap *neuron*, lalu hasilnya diberikan ke *layer* berikutnya.

Simbol  $\sigma$  merupakan fungsi aktivasi. *Output* dari *input layer* akan diterima sebagai *input* bagi *hidden layer*. Begitupun *hidden layer* akan mengirimkan hasilnya untuk *output layer*. Proses kegiatan ini dinamakan *feed forward*. *Multi layer perceptron* adalah gabungan dari banyak fungsi *non-linear*. pada *multi layer perceptron* masing–masing *neuron* saling terkoneksi dengan semua *neuron* pada *layer* berikutnya. Konfigurasi ini disebut sebagai *fully connected*. *Multi layer perceptron* secara umum menggunakan konfigurasi *fully connected*. Proses pembelajaran *multi layer perceptron* secara umum menggunakan *back-propagation* yaitu memperbaharui *parameter weight* pada setiap *synapse* secara bertahap dari *output layer* ke *input layer* berdasarkan nilai *error/loss* dari *output*.



Gambar 5 Multi layer perceptron

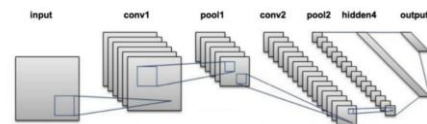
### 3. Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang menggambarkan hubungan antara tingkat aktivitas internal (*summation function*) yang mungkin berbentuk *linear* ataupun *non-linear*. befungsi untuk menentukan apakah *neuron* tersebut harus “aktif” atau tidak berdasarkan dari *weighted sum* (bobot dari hasil penjumlahan) dari *input*. Secara umum terdapat 2 jenis *activation function*, *linear* dan *non-linear activation function*. Secara umum, ada 3 *activation function* yang sering digunakan dalam jaringan *neural network*, diantaranya fungsi Sigmoid, Tanh, dan ReLu.

### G. Convolutional Neural Network

*Convolutional neural network* merupakan salah satu pengembangan dari jaringan syaraf tiruan yang terinspirasi dari jaringan syaraf manusia dan biasa digunakan pada data gambar untuk mendeteksi dan mengenali suatu objek pada sebuah gambar [5, hlm. 3]. *Convolutional neural network* adalah pengembangan dari *multilayer perceptron* yang didesain untuk mengolah data dua dimensi. pada *convolutional neural network*, setiap neuron direpresentasikan dalam bentuk dua dimensi, tidak seperti *multi layer perceptron* yang setiap *neuron* hanya berukuran satu dimensi. *Convolutional neural network* termasuk dalam *deep*

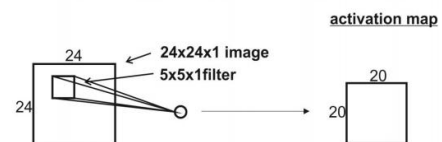
*neural network* karena memiliki kedalaman jaringan yang tinggi dan banyak untuk diaplikasikan pada data gambar. *Convolutional neural network* hampir sama dengan *neural network* pada umumnya yang memiliki *neuron* yang memiliki bobot dan bias dan fungsi aktivasi. Berikut merupakan alur proses *convolutional neural network*.



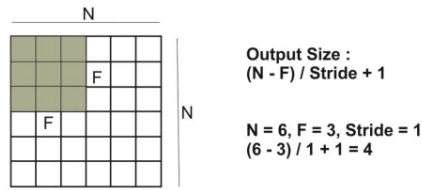
Gambar 6 Convolutional neural network

### 1. Convolution Layer

*Convolutional layer* merupakan bagian yang melakukan operasi konvolusi yaitu mengkombinasikan *linier filter* terhadap daerah lokal. *Layer* ini yang pertama kali menerima gambar yang diinputkan pada arsitektur. Bentuk *layer* ini adalah sebuah *filter* dengan panjang (*pixel*), tinggi (*pixel*), dan tebal sesuai dengan *channel image* data yang dimasukan. Ketiga *filter* ini akan bergeser keseluruhan bagian gambar. Pergeseran tersebut akan melakukan operasi “dot” antara *input* dan nilai dari *filter* tersebut sehingga akan menghasilkan *output* yang disebut sebagai *activation map* atau *feature map*. Pada gambar 2.9 menampilkan proses konvolusi yang ada di dalam *convolutional layer* dan Gambar 2.10 adalah cara menghitung nilai konvolusinya.



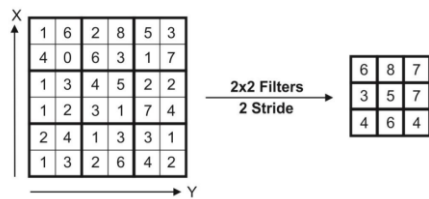
Gambar 7 Proses konvolusi pada convolution layer



Gambar 8 Cara menghitung nilai konvolusi

## 2. Pooling

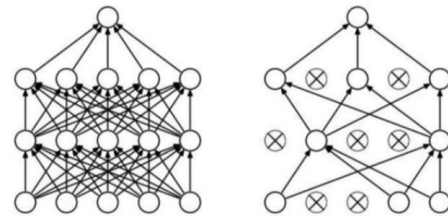
Pooling layer menerima *output* dari *convolutional layer*, pada *layer* ini ukuran data gambar akan direduksi. Prinsipnya *pooling layer* terdiri dari filter dengan ukuran tertentu dan *stride*, kemudian bergeser keseluruhan area feature map. Sebagian besar arsitektur *convolutional neural network*, metode *pooling* yang digunakan adalah *max pooling*. *Max pooling* membagi *output convolutional layer* menjadi beberapa *grid* kemudian setiap pergeseran *filter* akan mengambil nilai terbesar dari setiap *grid*. Tergantung pada panjang langkahnya, gambar yang dihasilkan adalah sebagian kecil dari ukuran aslinya yang berguna untuk mengurangi dimensi data, sehingga mengurangi jumlah parameter pada langkah selanjutnya. Pada gambar 2.9 menampilkan proses yang ada pada *pooling layer*:



Gambar 9 Proses yang ada pada pooling layer

## 3. Dropout

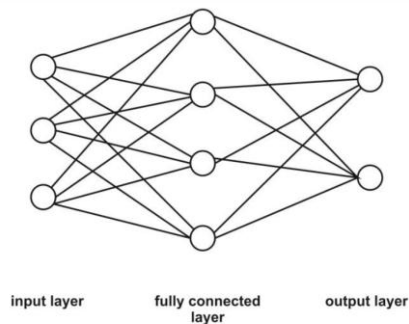
*Dropout* merupakan proses mencegah terjadinya *overfitting* dan juga mempercepat proses *learning*. *Dropout* menghilangkan *neuron* yang berupa *hidden* maupun *layer* yang *visible* di dalam jaringan. Dengan menghilangkan suatu *neuron*, berarti menghilangkannya sementara dari jaringan yang ada. *Neuron* yang akan dihilangkan akan dipilih secara acak. Setiap *neuron* akan diberikan *probabilitas* yang bernilai antara 0 dan 1 [6, hlm. 43].



Gambar 10 Contoh neural network yang menggunakan dropout dan tidak

## 3. Fully Connected

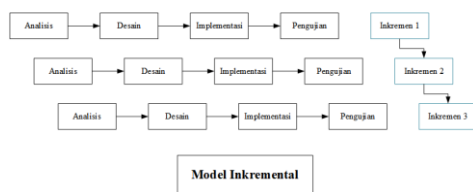
*Fully connected layer* mengambil *input* dari hasil *output pooling layer* yang berupa *feature map*. *Feature map* tersebut masih berbentuk *multidimensional array* maka lapisan ini akan melakukan *reshape feature map* dan menghasilkan vektor sebanyak *n*-dimensi dimana *n* adalah jumlah kelas *output* yang harus dipilih program. Misalnya lapisan terdiri dari 500 *neuron*, maka akan diterapkan *softmax* yang mengembalikan daftar *probabilitas* terbesar untuk masing-masing 10 label kelas sebagai klasifikasi akhir dari jaringan [7, hlm. 966]. Pada gambar 2.13 menampilkan proses yang ada dalam *fully connected layer*.



Gambar 11 Proses yang ada dalam fully connected layer

### H. Metode Inkremental

Metode inkremental adalah metode pengembangan perangkat lunak dengan tahapan analisis, desain, implementasi dan pengujian yang dilakukan secara urut untuk satu inkremennya, biasanya satu inkremen berfokus pada satu fitur. Model ini ditujukan untuk pengembangan perangkat lunak yang diperlukan fitur utamanya dirilis dalam waktu secepatnya, lalu kemudian diperbaiki atau menambah fitur nantinya pada inkremen selanjutnya [8, hlm. 43-44], Berikut merupakan ilustrasi metode pengembangan perangkat lunak inkremental.



Gambar 12 Model incremental

### I. Metode Pengujian Black Box

Metode pengujian *black box* adalah metode pengujian perangkat lunak yang memeriksa fungsionalitas aplikasi tanpa melihat proses yang terjadi didalamnya [9, hlm. 126]. Pengujian yang dilakukan berfokus untuk memenuhi kebutuhan yang telah ditetapkan sebelumnya dan fokus lainnya untuk menguji seberapa kuat sistem berjalan dengan beban proses yang berat.

### J. Metode Pengujian White Box

Metode pengujian *white box* adalah metode pengujian perangkat lunak yang bertujuan untuk memeriksa komponen program apakah berjalan semestinya dengan melihat internal kode program tersebut [10, hlm. 57]. Kelebihan metode pengujian *white box* adalah dapat mengetahui kesalahan logika, ketidaksesuaian asumsi, dan kesalahan ketik kode program dan kekurangan metode pengujian *white box* adalah memerlukan sumber daya yang besar untuk melakukannya.

### K. Python

*Python* adalah Bahasa pemrograman tingkat tinggi, yang dibuat oleh Guido van Rossum pada tahun 1980-an. Bahasa pemrograman *python* terstruktur dan berorientasi objek, Bahasa pemrograman ini juga dapat digunakan untuk banyak hal seperti pembuatan web, aplikasi desktop, analisis data dan kecerdasan buatan, [11, hlm. 8] dan pada penelitian ini digunakan untuk pembuatan model *convolutional neural network* beserta tampilannya.

### L. Tensorflow

*Tensorflow* adalah pustaka perangkat lunak yang berfokus pada pembuatan dan pelatihan pembelajaran mesin, *tensorflow* dikembangkan oleh tim google brain yang awalnya ditujukan untuk keperluan penelitian dan produksi google. *tensorflow* digunakan sebagai dasar dari keras dalam pembuatan *artificial neural network*, penggunaan *tensorflow* mempermudah pengembangan model pembelajaran mesin [12, hlm. 256-266].

### M. Keras

*Keras* adalah pustaka perangkat lunak yang menggunakan bahasa pemrograman *python*, keras berfungsi sebagai antarmuka *tensorflow* [13, hlm. 12]. *Keras* mendukung

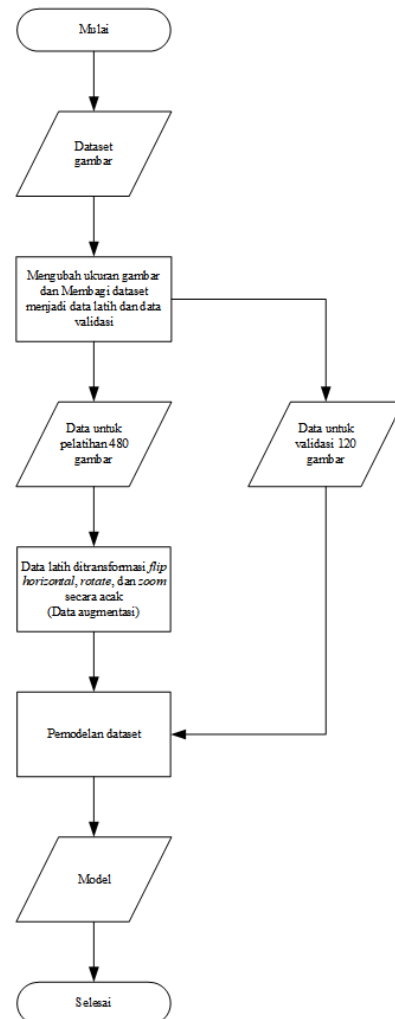
pengimplementasian *layers*, *activation function*, *optimizers* dalam pembuatan *neural network*. Dengan menggunakan pustaka perangkat lunak ini memungkinkan melakukan penelitian *neural network* dengan cepat

### III. HASIL DAN PEMBAHASAN

#### A. Inkremen Satu

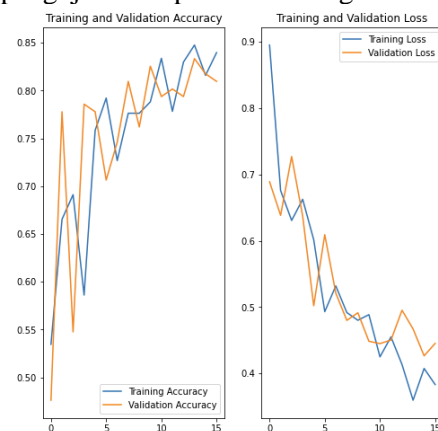
Tahap inkremen satu membahas tentang pembuatan model *convolution neural network* yang dapat melakukan prediksi sebagai inti perangkat lunak. Tahapan ini terdiri dari analisis, desain, implementasi dan pengujian.

Pada tahapan analisis dilakukan analisa metode, *dataset* dan *library* yang akan digunakan dalam pembuatan model, yaitu menggunakan 600 data gambar, 300 pada kategori berdoa dan 300 pada kategori khotbah, algoritma yang digunakan yaitu *convolutional neural network* dengan bantuan *library tensorflow keras* dan pada tahap desain dilakukan perancangan alur logika program sebagai berikut:



Gambar 13 Alur logika pada pembuatan model convolution neural network

Pada tahap implementasi dan pengujian didapati hasil sebagai berikut:



Gambar 14 Visualisasi hasil pelatihan model



Pada gambar diatas menunjukkan akurasi pada tahap validasi mencapai 83% dan tidak terjadi *overfitting*. Adapun percobaan yang dilakukan dengan melewati tahap data augmentasi, dan didapati akurasi 77% dan terjadi *overfitting*.



Gambar 14 Visualisasi hasil pelatihan model tanpa tahap data augmentasi dan dropout

### B. Inkremental Dua

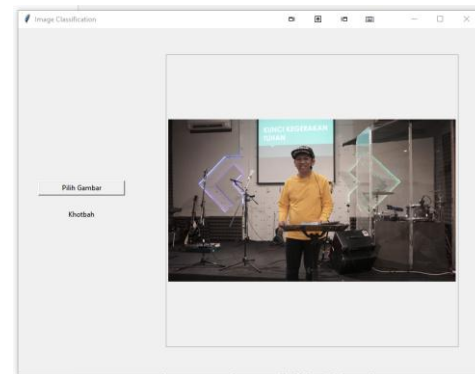
Pada tahap ini dilakukan analisis mengenai *library* yang akan digunakan untuk pembuatan *graphic user interface*, dan fungsi-fungsi yang dibutuhkan untuk menjalankan prediksi. Library yang digunakan adalah Tkinter yaitu pustaka perangkat lunak bawaan python untuk pembuatan *graphic user interface*. *Graphic user interface* dibuat berdasarkan *user requirement* sebagai berikut:

- Pengguna dapat memilih satu gambar
- Aplikasi dapat menampilkan gambar yang dipilih pengguna
- Aplikasi dapat menampilkan hasil prediksi

Elemen yang dibutuhkan adalah satu buah tombol untuk memilih gambar, bingkai untuk menampilkan gambar dan satu buah teks untuk menampilkan hasil prediksi. Fungsi yang dibutuhkan adalah fungsi tombol yang akan menyimpan *path* data gambar yang dipilih, agar dapat ditampilkan sekaligus

menjalankan fungsi prediksi yang telah dibuat pada inkremen satu dan menyimpan hasil prediksi pada sebuah *variable* yang nantinya akan ditampilkan.

Berikut merupakan tampilan program yang berhasil dibuat.



Gambar 15 Tampilan program

### IV. SIMPULAN

Berdasarkan hasil penelitian yang dilakukan untuk memprediksi kategori gambar kegiatan ibadah berdoa dan khotbah pada gereja SHOOT Fellowsip, maka dapat diambil kesimpulan sebagai berikut:

1. Metode Convolutional Neural Network mampu diterapkan untuk klasifikasi data gambar kegiatan ibadah gereja.
2. Penggunaan data augmentasi dan dropout pada model Convolution Neural Network dapat mengurangi overfitting pada model.
3. Hasil akurasi dari metode Convolutional Neural Network dengan menggunakan 600 gambar sebagai dataset dan dengan konfigurasi 16 epoch, 32 batch size adalah 83%.
4. Hasil akurasi yang didapatkan tanpa melalui tahap data augmentasi dan dropout adalah 77%.

**DAFTAR RUJUKAN**

- [1] Y. Sumarto, “Tinjauan Teologis Tentang Ibadah Bagi Pelaksanaan Misi Allah Theological Review of Worship For the implementation of God ’ s Mission,” *Jaffray*, vol. 17, no. 1, pp. 57–72, 2019, doi: 10.25278/jj.v17i1.312.
- [2] A. Kaplan and M. Haenlein, “Siri, Siri, in my hand: Who’s the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence,” *Bus. Horiz.*, vol. 62, no. 1, pp. 15–25, 2019, doi: 10.1016/j.bushor.2018.08.004.
- [3] T. Gonzalez and J. Diaz-Herrera, *Computing Handbook 3rd Edition Computer Science and Software Engineering*. 2014.
- [4] Bharadwaj, K. B. Prakash, and G. R. Kanagachidambaresan, *Pattern Recognition and Machine Learning*. 2021.
- [5] R. Nurfiti, “Implementasi Deep Learning Berbasis Tensorflow,” *J. Emit.*, vol. 18, no. 01, pp. 22–27, 2018.
- [6] R. Shanmugamani, *Deep Learning for Computer Vision*. 2018.
- [7] M. Z. Alom, P. Sidike, T. M. Taha, and V. K. Asari, “Handwritten Bangla Digit Recognition Using Deep Learning,” vol. 6, no. 7, pp. 990–997, 2017, [Online]. Available: <http://arxiv.org/abs/1705.02680>.
- [8] P. D. Roger S. Pressman and P. D. Bruce R. Maxim, *Software Engineering: A Practitioner’s Approach, Eighth Edition*. 2015.
- [9] J. Z. Gao, H.-S. J. Tsao, and Y. Wu, *Testing and Quality Assurance for Component-Based Software*. 2003.
- [10] Handy and J. Susilo, “Aplikasi Pengujian White-Box Ibbi Online Judge,” *J. Inform. dan Bisnis*, vol. 3, pp. 56–68, 2014.
- [11] J. Chan, *Learn Python in one day and learn it well: Python for beginners with hands-on project: the only book you need to start coding in Python immediately*. 2017.
- [12] J. D. Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, M. K. Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, P. T. Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, and G. B. Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng, “TensorFlow: A System for Large-Scale Machine Learning,” *Methods Enzymol.*, p. 21, 2016, doi: 10.1016/0076-6879(83)01039-3.
- [13] J. Moolayil, *Learn Keras for Deep Neural Networks: A Fast-Track Approach to Modern Deep Learning with Python*. 2019.