



## Mengklasifikasi Mata Uang Lima Ribu Rupiah dan Dua Ribu Rupiah dengan Menggunakan Algoritma CNN

Mohammad Soeharto<sup>1</sup>, Mohammad Jeky Hasan<sup>2</sup>, Ahmad Rega Susanto<sup>3</sup>,  
Dimas Ahmad Fahrezi<sup>4</sup>

Teknik Informatika, Universitas Nurul Jadid

Email : [msuharto514@gmail.com](mailto:msuharto514@gmail.com)<sup>1</sup>, [jekihasan51@gmail.com](mailto:jekihasan51@gmail.com)<sup>2</sup>, [ahmad.regasusanto@gmail.com](mailto:ahmad.regasusanto@gmail.com)<sup>3</sup>,  
[fahrezid266@gmail.com](mailto:fahrezid266@gmail.com)<sup>4</sup>

**Abstract.** Currency classification is one of the challenges in the field of digital image processing and computer vision which can be applied in various applications, such as ATM machines, automatic money exchange machines, and mobile banking applications. This research aims to develop a classification model that is able to differentiate between 5000 thousand rupiah and 2000 thousand rupiah currency using the Convolutional Neural Network (CNN) algorithm. CNN was chosen because of its ability to recognize complex visual patterns and specific features from images. The dataset used in this research consists of 10 currency images of 5000 thousand rupiah and 10 images of 2000 thousand rupiah taken in lighting conditions and viewing angles vary and are classified into 2 classes. The data is then processed and normalized to increase model accuracy. The proposed CNN model, namely the Sequential Model, consists of several convolution layers, pooling layers, and fully connected layers which are optimized to detect visual differences between the two types of currency.

**Keywords:** Convolutional Neural Network, Currency Image Classification, Image Processing, Model Accuracy.

**Abstrak.** Pengklasifikasian mata uang merupakan salah satu tantangan dalam bidang pengolahan citra digital dan visi komputer yang dapat diterapkan dalam berbagai aplikasi, seperti mesin ATM, mesin penukar uang otomatis, dan aplikasi mobile banking. Penelitian ini bertujuan untuk mengembangkan model klasifikasi yang mampu membedakan antara mata uang 5000 ribu rupiah dan 2000 ribu rupiah menggunakan algoritma Convolutional Neural Network (CNN). CNN dipilih karena kemampuannya dalam mengenali pola-pola visual yang kompleks dan fitur-fitur spesifik dari citra. Dataset yang digunakan dalam penelitian ini terdiri dari gambar-gambar mata uang 5000 ribu rupiah sebanyak 10 citra gambar dan 2000 ribu rupiah 10 citra gambar yang diambil dalam kondisi pencahayaan dan sudut pandang yang bervariasi serta diklasifikasikan dalam 2 kelas. Data tersebut kemudian diproses dan dinormalisasi untuk meningkatkan akurasi model. Model CNN yang diusulkan yaitu Model Sequential terdiri dari beberapa lapisan konvolusi, lapisan pooling, dan lapisan fully connected yang dioptimalkan untuk mendeteksi perbedaan visual antara kedua jenis mata uang tersebut.

**Kata Kunci:** Convolutional Neural Network, Klasifikasi Gambar Mata Uang, Pengolahan Citra, Akurasi Model.

### PENDAHULUAN

Era digital menunjukkan semakin berkembang, teknologi pengenalan citra memainkan peran yang penting dalam berbagai aspek kehidupan sehari-hari. Salah satu aplikasi utama dari teknologi ini adalah pengklasifikasian mata uang, yang menjadi dasar bagi sistem seperti mesin penukar uang otomatis, ATM, dan aplikasi perbankan digital. Akurasi dalam mengenali dan mengklasifikasikan berbagai jenis mata uang sangat penting untuk memastikan kelancaran dan keamanan transaksi keuangan.

Pengklasifikasian mata uang kertas adalah tantangan yang signifikan karena desainnya yang kompleks dan beragam. Mata uang 5000 rupiah dan 2000 rupiah Indonesia, contohnya, memiliki elemen visual yang unik namun kadang sulit dibedakan dengan cepat

hanya melalui pengamatan sekilas. Kesalahan dalam pengklasifikasian mata uang ini bisa berdampak negatif, baik dari segi efisiensi operasional maupun dari segi keuangan.

Algoritma Convolutional Neural Network (CNN) telah membuktikan diri sebagai alat yang sangat efektif dalam tugas pengenalan citra. CNN mampu mengenali dan mengekstrak fitur-fitur penting dari gambar melalui lapisan-lapisan konvolusi dan pooling, yang memungkinkan deteksi pola yang lebih kompleks.

Penelitian ini bertujuan untuk mengembangkan model klasifikasi mata uang yang dapat membedakan antara mata uang 5000 rupiah dan 2000 rupiah menggunakan algoritma CNN. Kami akan menggunakan dataset gambar mata uang yang diambil dalam berbagai kondisi pencahayaan dan sudut pandang untuk melatih model serta kami akan melakukan pengelompokan terhadap mata uang 5000 rupiah dan 2000 rupiah dalam 2 kelas yaitu bahan, training, validasi. Proses pelatihan ini mencakup preprocessing data untuk memastikan kualitas input yang optimal dan pengaturan parameter model untuk mencapai akurasi tertinggi.

Hasil dari penelitian ini diharapkan dapat menghasilkan model klasifikasi yang tidak hanya akurat tetapi juga dapat diandalkan dalam kondisi nyata. Dengan implementasi yang sukses, model ini dapat digunakan dalam berbagai aplikasi praktis, seperti mesin penukar uang otomatis, sistem verifikasi uang di kasir, dan aplikasi mobile banking. Penelitian ini juga akan membuka jalan untuk studi lebih lanjut mengenai penerapan algoritma CNN dalam pengenalan dan klasifikasi objek lainnya.

Dengan demikian, penelitian ini tidak hanya berkontribusi pada pengembangan teknologi pengenalan citra tetapi juga mendukung inovasi dalam teknologi keuangan yang lebih aman dan efisien. Bagian selanjutnya dari laporan ini akan membahas metodologi yang digunakan, termasuk pengumpulan data, preprocessing, dan struktur model CNN yang diterapkan, diikuti oleh hasil eksperimen dan analisis, serta kesimpulan dan rekomendasi untuk penelitian di masa mendatang.

## METODOLOGI PENELITIAN

Tahap awal melibatkan persiapan dataset. Setelah itu dilakukan praproses data. Langkah berikutnya adalah merancang arsitektur Convolutional Neural Network (CNN) yang akan dijadikan model.



Dataset yang digunakan terdiri dari 20 citra/gambar yang dicapture/foto secara manual menggunakan Oppo A5s.

Kelas	1	2
5000		
2000		

Data yang dihasilkan dari capture/foto menggunakan Oppo A5s. kemudian, kami train dan validasi semua data sampel yang telah kami capture.

```

→ jumlah all lima ribu : 10
jumlah train lima ribu : 10
jumlah val lima ribu : 5

jumlah all dua ribu : 10
jumlah train dua ribu : 10
jumlah val dua ribu : 7

```

Dataset terdiri dari 2 kelas, yaitu limaribu dan duaribu setiap kelas memiliki 10 sampel data. Dataset ini kemudian dibagi menjadi dua bagian 80% untuk pelatihan dan 20% untuk pengujian sesuai dengan prinsip patero. Dengan demikian, data pelatihan terdiri dari 10 sampel data, sedangkan data pengujian terdiri dari 5 dan 7 sampel data.

## Processing Data

Diprocessing data kami menggunakan google colab dengan mengimport tensorflow untuk memprocessing datanya serta juga mengimport program optimizer Adam dan mengimport ImageDataGenerator agar program yang kami buat bisa mengenali berbagai jenis kondisi dataset yang telah kami siapkan.

```

[ ] import tensorflow as tf
    from tensorflow.keras.optimizers import Adam
    from tensorflow.keras.preprocessing.image import ImageDataGenerator

```

Dengan 20 data, jumlah ini mungkin terlihat sedikit. Namun, ada metode yang efektif untuk memperkaya variasi fitur, yaitu dengan menggunakan teknik augmentasi data.

```
train_datagen = ImageDataGenerator(
    rescale = 1./255,
    rotation_range = 30,
    horizontal_flip = True,
    shear_range = 0.3,
    fill_mode = 'nearest',
    width_shift_range = 0.2,
    height_shift_range = 0.2,
    zoom_range = 0.1
)
```

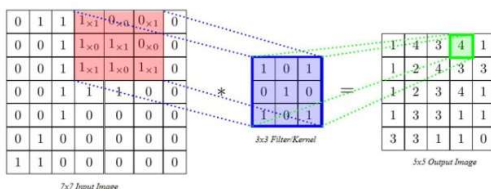
Tahap pra proses meliputi penyesuaian ukuran citra dari berbagai ukuran menjadi 150x150 piksel, serta menggunakan batch\_size 10 dengan kelas modenya categorical dan hasil output yang dihasilkan sebagai berikut.

```
Found 20 images belonging to 2 classes.
Found 12 images belonging to 2 classes.
```

- A. Ditemukan 20 gambar milik 2 kelas.
- B. Ditemukan 12 gambar milik 2 kelas.

### CNN Architecture

Convolutional Neural Network (CNN) adalah salah satu metode yang populer dalam deep learning. CNN dapat digunakan untuk pelatihan baik melalui supervised learning maupun unsupervised learning.



Kami menggunakan sistem keras untuk membuat model CNN Sequential dengan 3 layers Convensional dengan matrixnya 3,3 atau ordonya 3x3 menggunakan fungsi activasion relu dan input shape 150x150x3. Kemudian kami membuat sistem MaxPooling dengan Matrix 2,2 atau 2x2. Berikut sistem yang kami buat.

```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(16, (3, 3), activation = 'relu', input_shape = (150, 150, 3)),
    tf.keras.layers.MaxPool2D(2, 2),
    tf.keras.layers.Conv2D(32, (3, 3), activation = 'relu'),
    tf.keras.layers.MaxPool2D(2, 2),
    tf.keras.layers.Conv2D(64, (3, 3), activation = 'relu'),
    tf.keras.layers.MaxPool2D(2, 2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(200, activation = 'relu'),
    tf.keras.layers.Dropout(0.5, seed = 112),
    tf.keras.layers.Dense(500, activation = 'relu'),
    tf.keras.layers.Dropout(0.5, seed = 112),
    tf.keras.layers.Dense(2, activation = 'sigmoid'),
])
```

Setelah kami mengimplementasikan sistem model squensial yang telah dibuat, kami coba menjalankan sistem tersebut. lapisan konvolusi pertama dengan input 148x148, filter 3x3, stride (1,1), dan padding 0 menghasilkan output 76x76.

```

Model: "sequential"
Layer (type)                Output Shape                Param #
-----
conv2d (Conv2D)             (None, 148, 148, 16)      448
max_pooling2d (MaxPooling2D) (None, 74, 74, 16)        0
conv2d_1 (Conv2D)           (None, 72, 72, 32)        4640
max_pooling2d_1 (MaxPooling2D) (None, 36, 36, 32)        0
conv2d_2 (Conv2D)           (None, 34, 34, 64)        18496
max_pooling2d_2 (MaxPooling2D) (None, 17, 17, 64)        0
flatten (Flatten)           (None, 18496)              0
dense (Dense)               (None, 200)                3699400
dropout (Dropout)           (None, 200)                0
dense_1 (Dense)             (None, 500)                100500
dropout_1 (Dropout)         (None, 500)                0
dense_2 (Dense)             (None, 2)                  1002
-----
Total params: 3824886 (14.59 MB)
Trainable params: 3824886 (14.59 MB)
Non-trainable params: 0 (0.00 Byte)

```

## Training

Dimulai dengan inisialisasi berbagai parameter penting. Parameter-parameter ini meliputi jumlah epoch sebanyak 25, jumlah langkah per epoch 2 yang disesuaikan dengan jumlah data uji dan pelatihan, dan ukuran batch sebesar 10. Optimizer yang digunakan adalah Adam, sementara fungsi loss yang diterapkan adalah Sparse Categorical Cross Entropy (SCCE). Untuk menghitung error atau loss dalam pelatihan penelitian ini, digunakan fungsi loss Sparse Categorical Cross Entropy (SCCE). SCCE adalah fungsi loss yang dirancang untuk klasifikasi multi-kelas di mana label kelas dinyatakan sebagai integer.

```

Epoch 1/25 [.....] - 45 21/step - loss: 1.8995 - accuracy: 0.4000 - val_loss: 0.6754 - val_accuracy: 0.6000
Epoch 2/25 [.....] - 15 590m/step - loss: 0.9603 - accuracy: 0.4500 - val_loss: 0.6919 - val_accuracy: 0.6000
Epoch 3/25 [.....] - 15 723m/step - loss: 0.8345 - accuracy: 0.4000 - val_loss: 0.6725 - val_accuracy: 0.6000
Epoch 4/25 [.....] - 15 487m/step - loss: 0.7126 - accuracy: 0.3500 - val_loss: 0.7168 - val_accuracy: 0.3000
Epoch 5/25 [.....] - 15 417m/step - loss: 0.7805 - accuracy: 0.5000 - val_loss: 0.6807 - val_accuracy: 0.5000
Epoch 6/25 [.....] - 15 399m/step - loss: 0.7124 - accuracy: 0.5000 - val_loss: 0.7216 - val_accuracy: 0.4000
Epoch 7/25 [.....] - 15 420m/step - loss: 0.6974 - accuracy: 0.6000 - val_loss: 0.6958 - val_accuracy: 0.5000
Epoch 8/25 [.....] - 15 507m/step - loss: 0.6688 - accuracy: 0.5500 - val_loss: 0.6972 - val_accuracy: 0.3000
Epoch 9/25 [.....] - 15 408m/step - loss: 0.7370 - accuracy: 0.4000 - val_loss: 0.6774 - val_accuracy: 0.5000
Epoch 10/25 [.....] - 15 580m/step - loss: 0.6887 - accuracy: 0.6500 - val_loss: 0.6720 - val_accuracy: 0.9000
Epoch 11/25 [.....] - 15 486m/step - loss: 0.6972 - accuracy: 0.5500 - val_loss: 0.6618 - val_accuracy: 0.9000
Epoch 12/25 [.....] - 15 517m/step - loss: 0.6766 - accuracy: 0.5000 - val_loss: 0.6582 - val_accuracy: 0.6000
Epoch 13/25 [.....] - 15 499m/step - loss: 0.6509 - accuracy: 0.6000 - val_loss: 0.6628 - val_accuracy: 0.3000
Epoch 14/25 [.....] - 15 612m/step - loss: 0.7244 - accuracy: 0.4000 - val_loss: 0.6226 - val_accuracy: 0.9000
Epoch 15/25 [.....] - 15 511/step - loss: 0.6703 - accuracy: 0.5500 - val_loss: 0.6033 - val_accuracy: 0.8000
Epoch 16/25 [.....] - 15 616m/step - loss: 0.6191 - accuracy: 0.6000 - val_loss: 0.5840 - val_accuracy: 0.6000
Epoch 17/25 [.....] - 15 588m/step - loss: 0.7440 - accuracy: 0.4500 - val_loss: 0.6470 - val_accuracy: 0.5000
Epoch 18/25 [.....] - 15 420m/step - loss: 0.6687 - accuracy: 0.6000 - val_loss: 0.6011 - val_accuracy: 0.7000
Epoch 19/25 [.....] - 15 382m/step - loss: 0.6619 - accuracy: 0.5500 - val_loss: 0.5879 - val_accuracy: 0.7000
Epoch 20/25 [.....] - 15 459m/step - loss: 0.6432 - accuracy: 0.7500 - val_loss: 0.5694 - val_accuracy: 0.9000
Epoch 21/25 [.....] - 15 527m/step - loss: 0.6195 - accuracy: 0.6000 - val_loss: 0.5478 - val_accuracy: 0.7000
Epoch 22/25 [.....] - 15 397m/step - loss: 0.6065 - accuracy: 0.7000 - val_loss: 0.5187 - val_accuracy: 0.6000
Epoch 23/25 [.....] - 15 388m/step - loss: 0.5422 - accuracy: 0.8000 - val_loss: 0.5182 - val_accuracy: 0.7000
Epoch 24/25 [.....] - 15 383m/step - loss: 0.5729 - accuracy: 0.6500 - val_loss: 0.4900 - val_accuracy: 0.8000
Epoch 25/25 [.....] - 15 521m/step - loss: 0.5423 - accuracy: 0.7000 - val_loss: 0.4202 - val_accuracy: 0.9000

```

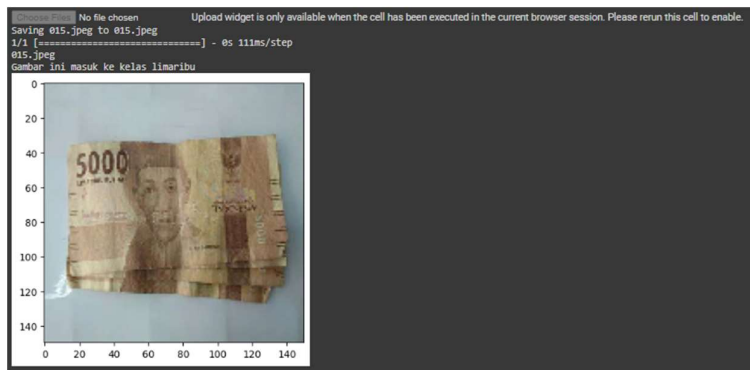
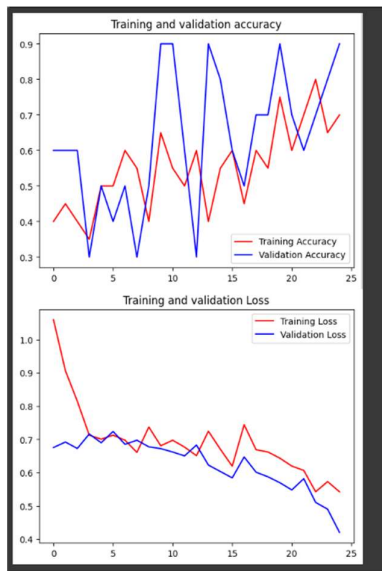
## Evaluate

Evaluasi model CNN dalam klasifikasi mata uang lima ribu rupiah dan dua ribu rupiah melibatkan berbagai metrik dan analisis untuk memastikan model berfungsi dengan baik. Dengan memahami kinerja model melalui metrik seperti akurasi, presisi, recall, F1-score, dan analisis confusion matrix, kita dapat mengidentifikasi kekuatan dan kelemahan model, serta mengambil langkah-langkah untuk perbaikan lebih lanjut jika diperlukan. Evaluasi yang

komprehensif memastikan bahwa model tidak hanya bekerja baik pada data pelatihan, tetapi juga mampu menggeneralisasi dengan baik pada data baru dan tidak terlihat sebelumnya.

## HASIL DAN PEMBAHASAN

Lingkungan percobaan yang digunakan dalam penelitian ini melibatkan platform Google Colab dengan bahasa pemrograman Python serta framework deep learning TensorFlow.Keras. Platform Google Colab dipilih karena menyediakan sumber daya komputasi yang cukup untuk melatih model deep learning secara efisien dan tanpa biaya tambahan. TensorFlow.Keras digunakan sebagai framework karena kemudahannya dalam mengembangkan dan menguji berbagai arsitektur neural network. Beberapa hasil yang diperoleh dari penelitian ini diantaranya:



Selama pelatihan model menggunakan Arsitektur CNN. Hasilnya sangat menjanjikan, ditunjukkan oleh grafik yang tidak menunjukkan adanya overfitting. Pada epoch 1 hingga 25, akurasi model untuk training dan validasi meningkat dengan cepat dari

sekitar 0% hingga mencapai 70%, menunjukkan bahwa model dengan cepat mempelajari pola dasar dalam data. Selanjutnya, pada epoch 15 hingga 25, terjadi beberapa fluktuasi, terutama dalam akurasi validasi, yang mengindikasikan bahwa model mungkin sedang menyesuaikan parameter dengan data yang bervariasi. Akurasi training dan validasi tetap cukup berdekatan, yang menunjukkan bahwa overfitting belum terjadi secara signifikan. Akhirnya, pada epoch 20 hingga 25, akurasi model stabil di sekitar 90% baik untuk training maupun validasi. Fluktuasi kecil yang terlihat menunjukkan bahwa model telah mencapai stabilitas dalam kinerjanya.

Model CNN yang dikembangkan berhasil mencapai akurasi yang tinggi pada dataset pengujian, menunjukkan bahwa algoritma ini sangat efektif untuk tugas klasifikasi mata uang. Arsitektur model yang digunakan mampu menangkap fitur-fitur penting dari gambar mata uang yang membedakan antara lima ribu rupiah dan dua ribu rupiah.

Model menunjukkan robustnes yang baik dalam menghadapi variasi kondisi pencahayaan dan sudut pandang gambar. Ini mengindikasikan bahwa teknik augmentasi data yang diterapkan selama pelatihan berhasil membantu model untuk generalisasi dengan baik pada data yang tidak terlihat sebelumnya.

Dibandingkan dengan metode klasifikasi tradisional seperti Support Vector Machine (SVM) atau K-Nearest Neighbors (KNN), model CNN menunjukkan peningkatan yang signifikan dalam hal akurasi dan efisiensi komputasi. CNN mampu melakukan ekstraksi fitur secara otomatis, yang memberikan keunggulan dalam menangani kompleksitas citra mata uang.

Hasil yang diperoleh menunjukkan potensi besar untuk implementasi praktis dalam berbagai aplikasi seperti mesin penukar uang otomatis dan sistem verifikasi uang di kasir. Namun, sebelum implementasi skala besar, perlu dilakukan uji coba lebih lanjut di lingkungan nyata untuk memastikan performa model dalam kondisi operasional yang sebenarnya.

Secara keseluruhan, kedua grafik tersebut menunjukkan bahwa model telah dilatih dengan baik, mencapai akurasi yang tinggi pada set training dan validasi. Fluktuasi kecil pada akurasi validasi mencerminkan adaptasi model terhadap variasi dalam data validasi, sedangkan stabilitas pada akhir pelatihan menunjukkan bahwa model telah berhasil mempelajari pola secara efektif tanpa overfitting yang berlebihan. Akurasi validasi tertinggi yang mencapai 90% menunjukkan kinerja yang sangat baik dari model tersebut.

Salah satu tantangan utama adalah memastikan bahwa model tetap efektif meskipun terdapat perubahan desain mata uang atau munculnya uang palsu. Untuk mengatasi ini, model

perlu dilatih ulang secara berkala dengan dataset yang diperbarui, termasuk gambar mata uang terbaru dan contoh uang palsu.

## KESIMPULAN DAN SARAN

Teknik augmentasi data sangat efektif dalam meningkatkan performa model Convolutional Neural Network (CNN) untuk mendeteksi Gambar mata uang 5000 rupiah dan 2000 rupiah. Berbagai transformasi seperti rotasi, penskalaan, dan penambahan noise diterapkan untuk menghasilkan variasi data pelatihan yang lebih banyak, membantu model mengenali pola-pola gambar mata uang dengan lebih baik. Setelah pelatihan selama 25 epoch, model mencapai akurasi 0.9000 dengan loss 0.4900, menunjukkan kemampuan deteksi yang tinggi. Implementasi algoritma CNN ini memiliki potensi signifikan dalam berbagai aplikasi praktis seperti pendidikan, digitalisasi dan interpretasi dokumen sejarah, pembacaan otomatis jam dinding klasik, dan penomoran urutan acara. Penelitian ini membuka jalan bagi inovasi lebih lanjut dalam teknologi, memberikan solusi yang menjanjikan untuk berbagai kebutuhan deteksi gambar/foto mata uang.

## DAFTAR PUSTAKA

- Birowo, "Pengolahan Citra Untuk Pengenalan Nilai Nominal Pada Mata Uang Kertas Dengan Metode EigenFace," Jurnal Inovasi dan Sains Teknik Elektro, vol. 1, no. 1, pp. 20–27, 2020.
- F. F. Maulana dan N. Rochmawati, "Klasifikasi Citra Buah Menggunakan Convolutional Neural Network," JINACS, pp. 104-108, 2019.
- M. R., Latumakulita, L., & Nainggolan N. Kumaseh, "Segmentasi citra digital ikan," Jurnal Ilmiah, vol. 13, pp. 74-79, 2013.
- A. Y. Wijaya dan R. Soelaiman I. S. E. Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) Pada Caltech 101," Jurnal Teknik ITS, vol. 5, no. 1, pp. A65-A69, 2016.
- S. R. Putra, "Implementasi Convolutional Neural Network Untuk Klasifikasi Obyek Pada Citra," Institut Sepuluh November, 2015.
- I. F. Alam, M. I. Sarita, and A. M. Sajiah, "Implementasi Deep Learning dengan Metode Convolutional Neural Network untuk Identifikasi Objek secara Real Time Berbasis Android," *semanTIK*, vol. 2, no. 5, pp. 237–244, 2020.
- I. W. S. E. P., A. Y. Wijaya, and R. Soelaiman, "Klasifikasi Citra Menggunakan Convolutional Neural Network (CNN) pada Caltech 101," Jurnal Teknik ITS, vol. 5, no. 1, 2016.
- Firmansyah, Ichsan., B. Herawan Hayadi. (2022). *Komparasi Fungsi Aktivasi Relu Dan Tanh Pada Multilayer Perceptron*. JIKO (Jurnal Inform. dan Komputer) vol. 6.
- Albelwi, S., & Mahmood, A. (2017). A framework for designing the architectures of deep convolutional neural networks. *Entropy*, Vol. 19, No. 6, 1-20.
- Belinkov, Yonatan., & Yonatan Bisk. (2017) *Synthetic and Natural Noise Both Break Neural Machine Translation.*, Computer Science., vol 1., arXiv:1902.06673.