

**UJIAN TENGAH SEMESTER**

**KLASIFIKASI DATA DINAMIS CITRA MENGGUNAKAN ALGORITMA  
*CONVOLUTIONAL NEURAL NETWORK***



**Dosen Pengampu:**

Desti Fitriati, S.Kom., M.Kom.

**Disusun Oleh:**

Faiqa Hadya Hanifa (4521210058)

Dheva Fauzia Chema (4521210066)

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS PANCASILA**

**JAKARTA**

**2024**

## Abstrak

Permainan batu, kertas, dan gunting merupakan permainan klasik yang sampai saat ini masih sering digunakan oleh berbagai kalangan masyarakat di seluruh dunia. Permainan ini sangat mudah untuk dimainkan karena hanya menggunakan tangan dari pemainnya sebagai alat untuk mewakili bentuk dari batu, kertas, dan gunting. Namun, bentuk tangan setiap orang tidak selalu sama, oleh karena itu diperlukan tahapan untuk mengklasifikasikan citra tangan untuk mendeteksi objek citra batu, kertas, dan gunting. Metode CNN digunakan untuk mengklasifikasikan data berlabel. Metode ini juga efektif dalam mengenali citra digital karena metode ini meniru cara kerja jaringan syaraf manusia dalam mengenali gambar visual. Pengujian ini bertujuan untuk menerapkan metode CNN dalam mengklasifikasikan citra tangan pada permainan batu, kertas, dan gunting. Tahapan proses klasifikasi CNN meliputi pengumpulan dataset, praproses data, pelatihan dan pengujian, pembangunan arsitektur model, serta pengujian akurasi. Berdasarkan hasil pengujian yang telah dilakukan, disimpulkan bahwa klasifikasi citra tangan untuk permainan batu, kertas, dan gunting menggunakan metode CNN memberikan hasil akurasi sebesar 99% pada data training dan akurasi sebesar 80% pada data validasi.

**Kata Kunci:** Klasifikasi, Permainan Batu Kertas Gunting, *Convolutional Neural Network*.

### 1. Pendahuluan

Terlibat dalam suatu permainan dapat menumbuhkan kreativitas serta mengasah pikiran karena memacu pemain untuk merasakan kemenangan atas permainan yang sedang mereka lakukan. Permainan klasik seperti batu, kertas, dan gunting merupakan permainan sederhana yang sampai saat ini masih populer di kalangan masyarakat seluruh dunia. Permainan ini hanya menggunakan tangan dari pemainnya sebagai alat untuk mewakili bentuk dari batu, kertas, dan gunting. Akan tetapi, pergerakan dan bentuk tangan setiap orang tidak selalu sama ketika memainkan permainan batu, kertas, dan gunting. Oleh karena itu, diperlukan suatu tahapan untuk mengklasifikasikan citra untuk mendeteksi objek gambar batu, kertas, dan gunting [2].

Terdapat banyak metode klasifikasi citra yang populer digunakan untuk mengklasifikasikan data citra diantaranya yaitu, *Convolutional Neural Network* (CNN), *Support Vector Machine* (SVM), *K-Nearest Neighbors* (KNN), *Backpropagation Neural Network*, dan metode lainnya yang termasuk pada lingkup pengolahan citra [3]. *Convolutional Neural Network* (CNN) merupakan metode *supervised learning* yang populer digunakan untuk mengklasifikasikan data citra yang berlabel. Metode ini memiliki konsep mirip dengan *Multi-Layer Perceptron* yang menggunakan algoritma *Backpropagation* dalam mengklasifikasi data, tetapi proses dari metode CNN didahului dengan proses pengenalan pola dari piksel suatu gambar sehingga meminimalkan *preprocessing* [4].

Berdasarkan uraian diatas, penulis menerapkan metode

*Convolutional Neural Network* (CNN) dalam proses klasifikasi pengenalan citra tangan dalam permainan batu, kertas, dan gunting. Adapun hasil yang diharapkan oleh penulis yaitu dapat memberikan pemahaman terhadap proses klasifikasi dan pengenalan citra tangan yang lebih baik dan akurat serta dapat menjadi referensi bagi pengembangan selanjutnya dalam lingkup topik pengolahan citra.

## 2. Landasan Teori

### 2.1 CNN dan Deep Learning

*Deep learning* merupakan salah satu cabang dari *machine learning* yang terdiri dari algoritma pemodelan abstraksi tingkat tinggi pada data menggunakan sekumpulan fungsi transformasi non-linier yang disusun secara berlapis dan mendalam. *Deep learning* memiliki perbedaan dengan *traditional machine learning* dalam hal tujuan pembelajaran, konstruksi model, dan pelatihan model. Selain itu, *deep learning* juga memiliki hierarki struktural tingkat yang lebih tinggi. *Deep learning* sendiri terdiri dari sejumlah jaringan saraf tiruan yang saling berhubungan, diantaranya yaitu CNN, RNN, LSTM, dan SOM. [5] Teknik dari *deep learning* ini dapat menghasilkan output yang berkualitas tinggi serta menghemat biaya operasional pada saat menangani data.

CNN atau *Convolutional Neural Network* merupakan bagian dari *deep learning* yang berperan sebagai pengenalan dan klasifikasi objek. CNN memiliki beberapa *layer* yaitu *input*

*layer*, *output layer*, dan *hidden layer* yang terdiri dari *convolutional layer*, *pooling layer*, dan *fully-connected layer* [6]. CNN telah berkembang sebagai pendekatan utama yang digunakan dalam proses klasifikasi berdasarkan informasi kontekstual. Metode ini merupakan metode pengembangan lebih lanjut dari metode MLP (*Multi Layer Perceptron*) [7] dimana *input layer* yang digunakan sebelumnya adalah bentuk dua dimensi, bukan berbentuk satu dimensi.

### 2.2 TensorFlow – Keras

TensorFlow merupakan generasi kedua dari sistem penelitian dan pengembangan kecerdasan buatan yang dikembangkan oleh Google sebagai pendukung metode CNN. *Tensor* merupakan unit data pusat dalam TensorFlow yang terdiri dari sekumpulan tipe data primitif yang dibentuk menjadi *array* dari dimensi apapun. Tipe data primitif *tensor* yaitu bilangan bulat, titik pecahan, karakter, *string* dan lain sebagainya. Dalam penggunaannya, TensorFlow memiliki satu *packages* yang dapat digunakan yaitu *Keras*. Keras digunakan untuk mengoptimalkan kinerja TensorFlow untuk mendapatkan hasil yang bagus secara cepat [8]. Keras mengadopsi prinsip pengungkapan progresif kompleksitas dimana alur kerja sederhana harus cepat dan mudah, sementara alur kerja lanjutan harus dimaksimalkan melalui jalur yang jelas dibangun berdasarkan apa yang telah dipelajari. Kinerja yang kuat dari Keras membuat metode ini banyak digunakan

oleh organisasi serta perusahaan termasuk NASA dan YouTube [9].

### 3. Pembahasan

#### 3.1 Metode

Penulis memiliki tujuan untuk membuat model pembelajaran mendalam untuk mengenali data citra tangan yang menyerupai bentuk batu, kertas, dan gunting secara akurat. Proses ini menggabungkan beberapa lapisan dengan metode CNN untuk meningkatkan akurasi klasifikasi. Digunakan jaringan CNN dengan 2 lapisan konvolusi dan 2 lapisan *dense*. Dalam 2 lapisan di mana lapisan konvolusi diikuti oleh aktivasi *Rectified Linear Unit Layer* (ReLU) dan lapisan *MaxPool*. Lapisan *dense* pertama diikuti aktivasi ReLU dan lapisan terakhir softmax untuk mendapatkan probabilitas setiap kelas.

$$ReLU, f(x) = \max(0, x) \quad [1]$$

Dimana  $x$  adalah nilai input ke fungsi. Optimalisasi jaringan melalui ReLU menukar setiap nilai negatif dari *pooling layer* dengan 0. Langkah ini membantu proses CNN tetap stabil secara matematis dan menjaga nilai yang telah dipelajari untuk tidak 0 atau menuju tak terhingga. *Fully-connected layer* membutuhkan aktivasi *softmax* pada lapisan tahap akhir untuk proses kompilasi model yang efektif. Fungsi aktivasi ini memiliki masukan  $x$  seperti persamaan berikut:

$$\sigma(x) = \frac{1}{(1+e^{-x})}$$

#### 3.2 Dataset

Data yang digunakan diambil dengan menarik dari tautan umum yang kemudian menyimpannya ke dalam direktori tempat *notebook* dijalankan. Dataset merupakan citra tangan menyerupai batu, kertas, dan gunting. Data terdiri dari 2520 gambar yang terbagi menjadi tiga kelas berbeda: *rock*, *paper*, dan *scissors*. Masing-masing kelas tersebut berjumlah 840 data citra dalam bentuk .png. Himpunan data ini akan digunakan untuk proses pelatihan dan validasi.

```
Total training rock images 840
First 10 Rock files : ['rock06ck02-071.png', 'rock04-116.png', 'rock03-06
Total training Paper images 840
First 10 Paper files : ['paper06-050.png', 'paper03-056.png', 'paper01-02
Total training Scissors images 840
First 10 Scissors files : ['testscissors01-105.png', 'scissors01-053.png']
```

Gambar 1. Rincian Dataset pada tiap Kelas



Gambar 2. Sampel dari Dataset RPS

##### 3.2.1 Split Dataset

Untuk melatih dan mengevaluasi sistem klasifikasi yang dibangun, dataset dibagi menjadi dua kategori *training* dan *validation* dengan rasio 80:20. Data latih berisi 2016 data dan sisa 504 data termasuk data validasi. Pembagian data diatur menggunakan parameter *validation\_split* dari fungsi *ImageDataGenerator* yang

disediakan dalam modul *preprocessing* gambar oleh Keras.

```
Found 2016 images belonging to 3 classes
Found 504 images belonging to 3 classes
```

Gambar 3. Pemisahan Dataset

Dari modul yang sama, gambar dari dataset dapat diperbanyak variasinya dengan melakukan proses augmentasi, seperti normalisasi, rotasi, putar balik, pangkas, perbesar, dan pengisian pixel, sehingga meningkatkan akurasi model dalam memahami gambar yang beragam.

```
training_datagen = ImageDataGenerator(
    rescale=1. / 255,
    rotation_range=40,
    horizontal_flip=True,
    shear_range=0.2,
    zoom_range=0.2,
    fill_mode='nearest',
    validation_split=0.2
)
```

Gambar 4. *Image Preprocessing*

### 3.2.2 Preprocessing

Praproses data dilakukan sebelum proses pelatihan. Proses ini bertujuan untuk mempersiapkan data mentah menjadi data yang dapat digunakan ketika memasuki lapisan jaringan syaraf buatan, termasuk menetapkan label pada data.

```
train_generator = training_datagen.flow_from_directory(BASE_DIR,
                                                        class_mode='categorical',
                                                        batch_size=32,
                                                        target_size=(224, 224),
                                                        subset='training')

val_generator = training_datagen.flow_from_directory(BASE_DIR,
                                                      class_mode='categorical',
                                                      batch_size=32,
                                                      target_size=(224, 224),
                                                      subset='validation')
```

Gambar 5. *Data Preprocessing*

## 3.3 Model

Dapat dilihat pada gambar 6 bahwa pembangunan arsitektur model menggunakan model *Sequential* dengan basis algoritma CNN. *Convolution layer* berfungsi untuk menemukan fitur-fitur dari data citra dengan menggunakan filter (kernel) yang digunakan dan menghitung kecocokan fitur tersebut selama pengujian.

```
model = tf.keras.models.Sequential([
    # The input shape is the desired size of the image with 3 bytes color of rgb
    # First convolution layer
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPool2D(2, 2),
    # Second convolution layer
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPool2D(2, 2),
    # Flatten the results to feed into a DNN
    tf.keras.layers.Flatten(),
    # 512 neuron hidden layer
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(3, activation='softmax')
])
```

Gambar 6. Rincian Lapisan Syaraf Tiruan

Berdasarkan gambar 7, lapisan *Input Layer* dari barisan CNN digunakan sebagai lapisan masukan yang mengambil gambar, kemudian mengubah ukurannya dan diteruskan ke lapisan selanjutnya untuk ekstraksi fitur. Lapisan pertama ini memiliki 32 bentuk keluaran dengan 3 dimensi filter untuk setiap langkah dan lapisan masukan 150x150 dengan 3 byte warna. Kemudian, untuk fungsi aktivasi, ditambahkan ReLU. Fitur yang telah diekstraksi diteruskan ke dalam *MaxPooling Layer* dengan ukuran 2x2. *MaxPooling* menguraikan informasi dan mengurangi dimensi data dengan menyimpan nilai maksimum dari tiap kelompok pixel pada gambar sementara mempertahankan fitur-fiturnya terbaik.

Lapisan ini digunakan untuk mempercepat komputasi, memudahkan identifikasi, dan mengurangi *overfitting*

[10]. Hal ini diulangi lagi pada lapisan berikutnya dengan menaikkan nilai filter dari 32 menjadi 64. *Flatten layer* digunakan untuk diteruskan ke dalam lapisan *fully-connected layer* atau lapisan penghubung sepenuhnya [11]. Lapisan *dense* mengambil fitur tingkat tinggi dari filter gambar dan menerjemahkannya ke dalam kategori dengan label yang telah ditentukan. Lapisan tersembunyi dikompres kedalam *layer dense* pertama menggunakan aktivasi ReLU dengan 512 neuron dan aktivasi *softmax* pada *layer output* dengan 3 node keluaran.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 148, 148, 32)	896
max_pooling2d (MaxPooling2D)	(None, 74, 74, 32)	0
conv2d_1 (Conv2D)	(None, 72, 72, 64)	18496
max_pooling2d_1 (MaxPooling2D)	(None, 36, 36, 64)	0
flatten (Flatten)	(None, 82944)	0
dense (Dense)	(None, 512)	4246784
dense_1 (Dense)	(None, 3)	1539

-----  
Total params: 42488771 (162.08 MB)  
Trainable params: 42488771 (162.08 MB)  
Non-trainable params: 0 (0.00 Byte)

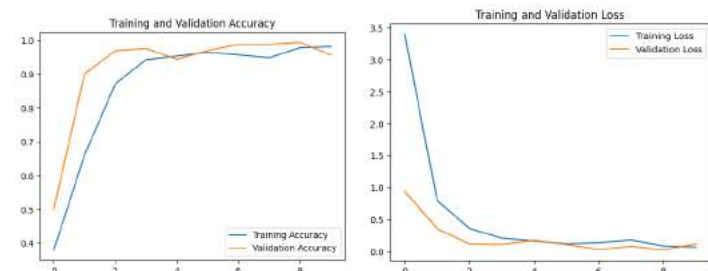
Gambar 7. *Hyperparameter Model CNN*

Setelah model dibangun, parameter dalam data pelatihan yang dirinci pada gambar sebelumnya memasuki proses kompilasi dengan menetapkan konfigurasi *loss*, *optimizer*, dan *metric*. Metode optimasi dapat diubah sesuai dengan hasil eksperimen yang dicoba dan bergantung dengan kebutuhan, seperti ukuran dataset, sensitivitas pada momentum, dan ukuran komputasi. Dalam penelitian ini, RMSprop digunakan sebagai metode

optimasi karena penerapannya yang stabil dalam mengatasi fluktuasi, kecepatan pembelajaran efektif, dan kemampuan konvergensi yang cepat. Kemudian, *categorical\_crossentropy* sebagai fungsi kerugian parameter data sesuai dengan tipe dataset yang memiliki banyak pola atau *multivariate*.

### 3.4 Hasil

Setelah melakukan iterasi *training*, akurasi dan kerugian atau *loss* dari proses tersebut divisualisasikan menggunakan grafik untuk membandingkan perubahan yang terjadi saat pelatihan dan validasi dan menganalisis performa model. Gambar 8 mengilustrasikan hasil akurasi dan kerugian untuk setiap epoch.



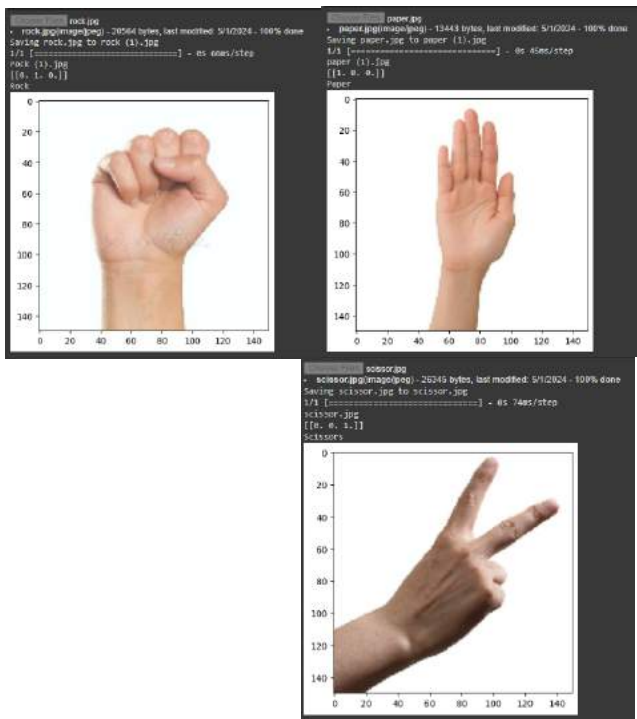
Gambar 8. *Accuracy Plot dan Loss Plot*

Proses klasifikasi CNN dari 80% himpunan 2016 data *training* dan 20% himpunan 504 data *validation* menunjukkan peningkatan akurasi dan penurunan kerugian. Akurasi proses pelatihan mencapai 99% dan kerugian

Peningkatan tersebut disebabkan oleh penggunaan nilai epoch yang tinggi dan jumlah data tiap iterasi yang diatur. Selama proses penelitian, nilai *epoch* yang digunakan beragam untuk mendapatkan jumlah iterasi yang tepat. Model cenderung mengalami *overfitting*

ketika iterasi pelatihan dibawah 20 dan memprediksi kelas yang keliru. Selain itu, peningkatan terjadi sebagai hasil dari pengembangan hiperparameter CNN yang diterapkan lebih menyeluruh; proses penyaringan pada lapisan *MaxPooling* dan pemilihan metode optimasi yang mengurangi *overfitting* dan mengoptimalkan akurasi.

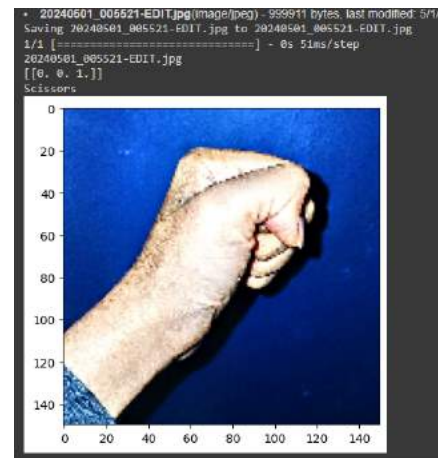
Pengecekan dan evaluasi model dilakukan dengan memprediksi gambar baru melalui unggahan. Dapat dilihat pada Gambar 9 bahwa data citra yang baru terklasifikasi kedalam kelasnya dengan benar. Dengan menggunakan nilai yang paling akurat dalam perbandingan tiga neuron output, nilai yang diantisipasi 0 adalah kertas, 1 adalah batu, dan oleh karena itu untuk objek lainnya menyarankan bahwa gambar tersebut adalah gunting.



Gambar 9. Hasil Prediksi Gambar Baru

#### 4. Eksperimen dan Diskusi

Hasil penelitian yang telah dilakukan menggambarkan arsitektur yang ideal dari klasifikasi dataset gunting, batu, kertas. Namun, selama proses percobaan penulis melakukan beberapa tahap eksperimen untuk menghindari *overfitting* pada citra baru dan meningkatkan stabilitas saat konvergensi proses validasi. Dapat dilihat pada gambar 10 bahwa model memprediksi kelas citra yang salah ketika menerima objek gambar yang lebih kompleks daripada dataset.



Gambar 10. Hasil Prediksi Salah

Maka, dilakukan eksperimen dengan menambahkan lapisan konvolusi, *dropout*, dan *tuning* terhadap parameter pada *fully connected layer*. Metode kerugian dan jumlah filter juga disesuaikan untuk mendapatkan grafik akurasi yang baik dan stabil. Gambar 11 memperlihatkan percobaan pada model yang menghasilkan grafik performa pada gambar 12.

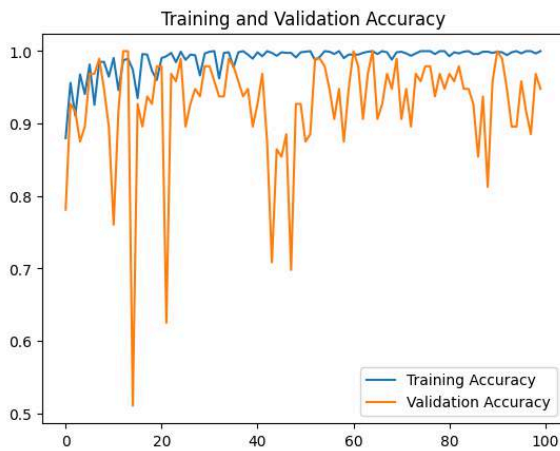


```

model = tf.keras.models.Sequential([
    # The input shape is the desired size of the image with 3 bytes color of r
    # First convolution layer
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPool2D(2, 2),
    # Second convolution layer
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPool2D(2, 2),
    # Third convolution layer == add
    tf.keras.layers.Conv2D(64, (3, 3), activation='relu'),
    tf.keras.layers.MaxPool2D(2, 2),
    # Flatten the results to feed into a DNN
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dropout(0.3), # add
    # 512 neuron hidden layer
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(128, activation='relu'), # add
    tf.keras.layers.Dense(3, activation='softmax')
])
model.summary()

```

Gambar 11. Arsitektur Model Percobaan



Gambar 12. Grafik Hasil Akurasi

Jika dibandingkan dengan penelitian sebelumnya, model CNN yang dibangun dan jumlah *epoch* yang digunakan. Walaupun, stabilitas tidak dicapai pada proses pengujian dengan jumlah iterasi pembelajaran 100, akurasi akhir menyentuh 95%.

```

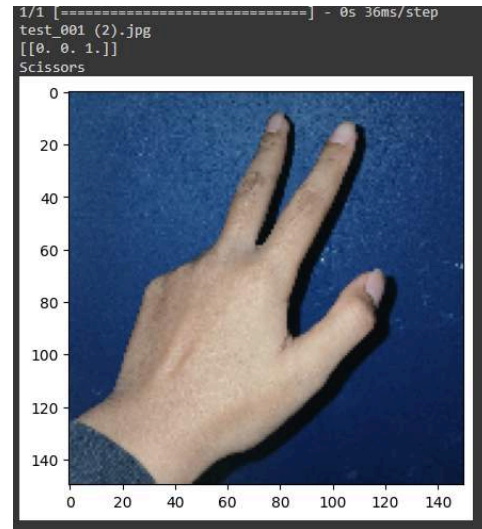
Accuracy reached: [1.0]
Validation Accuracy reached: [0.9479166865348816]
Loss reached: [0.00010062672663480043]
Validation Loss reached: [0.1468721479177475]

val_loss, val_accuracy = model.evaluate(val_generator, verbose=2)

16/16 - 9s - loss: 0.1824 - accuracy: 0.9623 - 9s/epoch - 577ms/st

```

Gambar 13. Akurasi Pengujian Model Percobaan



Gambar 13. Prediksi Gambar Model Percobaan

Eksperimen terus dilakukan hingga model dapat memprediksi data citra baru dengan akurat. Penelitian pada dataset menggunakan algoritma CNN disesuaikan dengan kebutuhan dan tujuan hasil akhir yang diinginkan. Maka, akan menghasilkan arsitektur jaringan syaraf tiruan yang berbeda-beda.

## 5. Penutup

Berdasarkan dari hasil pengujian yang telah dilakukan, dapat disimpulkan bahwa klasifikasi citra tangan untuk permainan batu, kertas, dan gunting menggunakan metode CNN (*Convolutional Neural Network*) yang dioptimalkan dengan kinerja TensorFlow memberikan hasil akurasi yang cukup baik. Model pembelajaran mendalam yang telah dibuat berhasil menerapkan metode CNN sebagai metode klasifikasi dengan tingkat akurasi pada data training sebesar 99% serta akurasi pada data validasi sebesar 80%. Akurasi yang tinggi ini mengindikasikan keunggulan model dalam klasifikasi gambar gunting, batu, dan kertas dengan akurat.



## Referensi

- [1] Y. Zhang, I. R. Porter, M. Wieland, and P. S. Basran. Sepa-rable confident transductive learning for dairy cows teat-end condition classification. *Animals*, 12(7):886, 2022
- [2] Naufal, M. F., Huda, S., Budilaksono, A., Yustisia, W. A., Arius, A. A., Miranti, F. A., & Prayoga, F. A. T. (2021). Klasifikasi Citra Game Batu Kertas Gunting Menggunakan Convolutional Neural Network. *Techno. Com: Jurnal Teknologi Informasi*, 20(1), 166-174.
- [3] Darugutni, A., & Marcos, H. (2023). KLASIFIKASI PERMAINAN BATU KERTAS GUNTING MENGGUNAKAN ALGORITMA CONVOLUTION NEURAL NETWORK (CNN). *METHODIKA: Jurnal Teknik Informatika dan Sistem Informasi*, 9(1), 1-3.
- [4] Wijaya, K., & Widiyanto, E. P. (2023, April). Klasifikasi Kepemilikan Tanda Tangan Menggunakan Convolutional Neural Network dengan Arsitektur AlexNet. In *MDP Student Conference* (Vol. 2, No. 1, pp. 133-143).
- [5] Ichsan, M. N., Armita, N., Minarno, A. E., & Sumadi, F. D. S. (2022). Increased Accuracy on Image Classification of Game Rock Paper Scissors using CNN. *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, 6(4), 606-611.
- [6] ABDULFATTAH, M. E., NOVAMIZANTI, L., & RIZAL, S. (2021). Super Resolution pada Citra Udara menggunakan Convolutional Neural Network. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, 9(1), 71.
- [7] Salawazo, V. M. P., Gea, D. P. J., Gea, R. F., & Azmi, F. (2019). Implementasi metode convolutional neural network (cnn) pada peneganaan objek video cctv. *Jurnal Mantik Penusa*, 3(1.1).
- [8] Tiku, J. C., Saputra, W. A., & Prasetyo, N. A. (2022). Pengembangan Sistem Deteksi Memakai Masker Menggunakan Open CV, Tensorflow dan Keras. *JURIKOM (Jurnal Riset Komputer)*, 9(4), 1183-1190.
- [9] Mujilahwati, S., Sholihin, M., & Wardhani, R. (2021). Optimasi Hyperparameter TensorFlow dengan Menggunakan Optuna di Python: Study Kasus Klasifikasi Dokumen Abstrak Skripsi. *Jurnal Media Informatika Budidarma*, 5(3), 1084-1089.
- [10] Y. A. Putri, E. C. Djamal, and R. Ilyas, "Identification of Medicinal Plant Leaves Using Convolutional Neural Network," *J. Phys. Conf. Ser.*, vol. 1845, no. 1, 2021, doi: 10.1088/1742-6596/1845/1/012026.
- [11] H. Fonda, Y. Irawan, A. Febriani, S. Informatika, and H. T. Pekanbaru, "Klasifikasi Batik Riau Dengan Menggunakan Convolutional Neural Networks (Cnn): Klasifikasi Batik Riau Dengan Menggunakan Convolutional Neural Networks," *jik.http.ac.id*, vol. 9, no. 1, pp. 7–10, 2020. Available: <https://jik.http.ac.id/index.php/jik/article/view/14>