



## **PENERAPAN ALGORITMA CNN (CONVOLUTIONAL NEURAL NETWORK) UNTUK DETEKSI DAN KLASIFIKASI TARGET MILITER BERDASARKAN CITRA SATELIT**

**Muhamad Luthfi Bangun Permadi, Restu Gumilang**

Sekolah Tinggi Teknologi Bandung, Indonesia

Email: luthfinewzoneonline@gmail.com, estu21240@gmail.com

### **Abstrak**

Situasi geopolitik dunia saat ini cenderung kurang baik, dimana terdapat sebuah ketegangan di beberapa wilayah regional, mulai dari eropa, hingga indo-pasifik. Bahkan konfrontasi militer tak bisa dihindarkan, bukan tidak mungkin, pecah perang yang lebih besar, dengan begitu kesiapan Indonesia dipertanyakan. Sebagai negara yang luas, wilayah Indonesia rentan terhadap serangan. Dengan kekuatan militer dan dukungan teknologi yang lemah tentu hal yang buruk. Dalam perang, target yang menjadi sasaran utama adalah sistem pertahanan. Disini kami mencoba menerapkan algoritma CNN untuk mendeteksi sekaligus mengklasifikasikan bangunan fasilitas militer, alutsista, dan objek yang bukan merupakan militer. Tujuannya adalah model deep learning dapat membantu peralatan tempur dalam menargetkan sasaran secara efektif, melalui citra satelit. CNN yang diuji menggunakan beberapa model, dan optimizer yang berbeda, untuk mengetahui model mana yang memiliki akurasi terbaik. Model diberikan data gambar dalam tiga kelas, kemudian dilakukan konvolusi dan max pooling melalui layer-layernya, hingga melalui 512 neuron di tahap klasifikasi dan output di 3 neuron sesuai jumlah kelas. Hasilnya adalah, model terbaik dengan menggunakan arsitektur InceptionV3 dan Adamax optimizer, menghasilkan nilai validasi akurasi sebesar 96% dan loss validation-nya 0,1757. Serta classification report dengan masing-masing presisi kelas defense equipment sebesar 100%, military facilities sebesar 92%, dan non-military sebesar 94%.

**Kata kunci:** CNN, arsitektur, klasifikasi

### **Abstract**

*The current geopolitical situation in the world tends to be unfavorable, where there is tension in several regional areas, from Europe to the Indo-Pacific. In fact, military confrontation cannot be avoided, it is not impossible that a bigger war will break out, so that Indonesia's readiness is questioned. As a large country, Indonesia's territory is vulnerable to attacks. With weak military strength and technological support, this is certainly a bad thing. In war, the main target is the defense system. Here we try to apply the CNN algorithm to detect and classify military facilities, defense equipment and non-military objects. The goal is that deep learning models can help combat equipment target targets effectively, through satellite imagery. The CNN was tested using several models, and different optimizers, to find out which model had the best accuracy. The model is given image data in three classes, then convolution and max pooling are carried out through its layers, up to 512 neurons in the classification stage and output in 3 neurons according to the number of classes. The result is, the best model using InceptionV3 architecture and Adamax optimizer, produces an accuracy validation value of 96% and the validation loss is 0,1757. As well as a classification report with each class of defense equipment precision of 100%, military facilities of 92%, and non-military of 94%.*

**Keywords:** CNN, architecture, classification

### **PENDAHULUAN**

Situasi geopolitik global yang saat ini sedang tidak dalam arah yang baik, mendorong eskalasi konflik yang kian meluas, dan hal itu sudah menjadi perhatian penting bagi sejumlah pihak. Dalam sebuah konflik antar negara, tentu militer merupakan opsi terakhir yang dijadikan solusi untuk menghadapi ketegangan, namun bukan tidak mungkin kekuatan militer pada akhirnya digunakan. Di era industri 4.0 sekarang, konflik tidak lagi hanya melibatkan unsur militer, namun terdapat unsur teknologi informasi yang dapat menentukan kemenangan.

Dengan kecanggihan teknologi saat ini, alutsista tua tidak dapat lagi diandalkan untuk menghadapi modern warfare. Teknologi harus diatasi dengan teknologi. Posisi pertahanan merupakan

hal yang penting dalam konflik. Terkadang sejumlah trik digunakan untuk mengelabui sensor deteksi, yang menyebabkan semakin sulitnya mengarahkan senjata pada lawan. Ketika target strategis disamarkan dengan lingkungan sekitar, maka radar dapat digunakan sebagai solusi, namun jika target strategis mengenakan sistem anti-radar serta target dalam posisi terbuka, maka dengan penginderaan jarak jauh dapat digunakan sebagai solusi.

Teknologi satelit adalah salah satu yang menjadi andalan dalam konflik. Satelit dapat berperan sebagai alat penginderaan jarak jauh posisi target strategis. Dalam penginderaan jarak jauh tersebut, kami mencoba menerapkan algoritma kecerdasan buatan CNN (Convolutional Neural Network). CNN adalah algoritma deep learning yang dapat dilatih untuk kumpulan data gambar skala besar untuk melakukan pengenalan dan deteksi objek dalam gambar (Ayu, Kinasih, & Hidayat, 2021a). Penelitian ini diharapkan menghasilkan sebuah model yang berperan untuk mendeteksi dan mengklasifikasi target strategis melalui deep learning atau pembelajaran mendalam citra satelit. Tujuannya, dengan bantuan model CNN, posisi dan letak target strategis lawan dapat terbuka secara visual, sehingga diharapkan dapat ditindak lebih lanjut oleh sistem pendukung pertahanan yang terintegrasi dengan satelit. Model juga sekaligus mengatasi masalah persenjataan yang dilengkapi sistem anti-radar, sehingga target strategis lawan tetap diketahui posisinya.

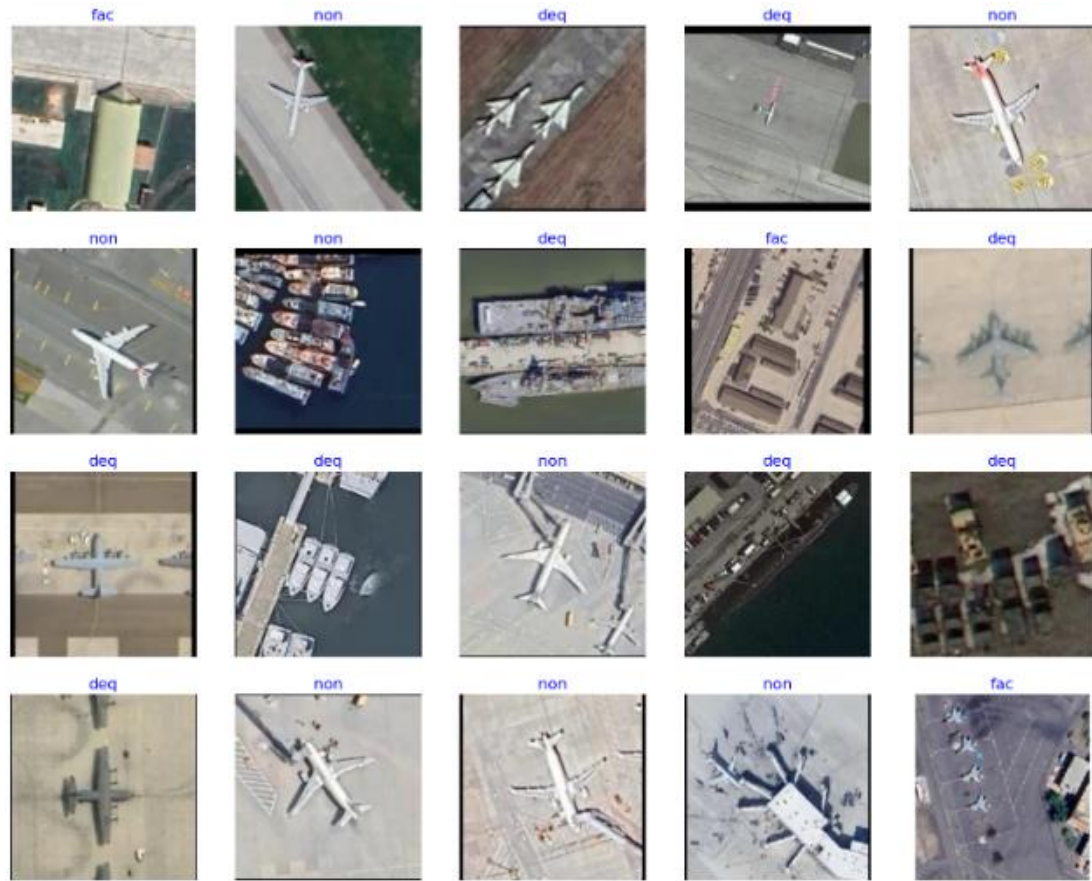
Pada penelitian sebelumnya, CNN digunakan untuk penginderaan jarak jauh menggunakan satelit SPOT6 untuk mengklasifikasi citra satelit tutupan lahan dengan hasil terbaik menggunakan Adam optimizer, learning rate 0.001 dan epoch 100 (#1 dkk., t.t.). Hasilnya, model memiliki akurasi 95,45 %, dan loss 0,2457. Kemudian pada penelitian (Ayu, Kinasih, & Hidayat, 2021b), penginderaan jarak jauh mengambil citra gambar menggunakan UAV (Unmanned Aerial Vehicle) untuk melakukan pemetaan persebaran bangunan di desa Campurejo menggunakan Mask R-CNN. Hasilnya untuk wilayah I mendapat nilai presisi sebesar 94,78%, sedangkan wilayah II sebesar 98,10%. Kemudian CNN juga digunakan dalam penelitian (Miranda & Aryuni, t.t.) untuk klasifikasi tutupan lahan menggunakan citra satelit Sentinel-2, hasilnya adalah segmentasi 2072 objek menggunakan eCognition dengan akurasi model 98,4%. Pada penelitian (Hermawan E, 2023), CNN digunakan untuk deteksi penyakit tanaman padi dengan menggunakan UAV. Hasilnya pada percobaan terakhir, didapat akurasi model 95%.

### **METODE PENELITIAN**

Untuk data pelatihan model, kami menggunakan tiga kelas klasifikasi data citra satelit, yakni alutsista (*defence equipments*) disingkat *deq*, fasilitas militer (*military facilities*) disingkat *fac*, dan objek non-militer (*non-militaries*) disingkat *non*. Alutsista untuk target strategis bernilai rendah, fasilitas militer untuk target strategis bernilai tinggi, serta objek non-militer sebagai data negatif, atau objek yang dihindari dari penargetan. Data pelatihan diambil dari citra satelit *landsat-8* dalam format jpg. Dimensi citra data pelatihan dibuat seragam yakni 150 x 150 piksel.

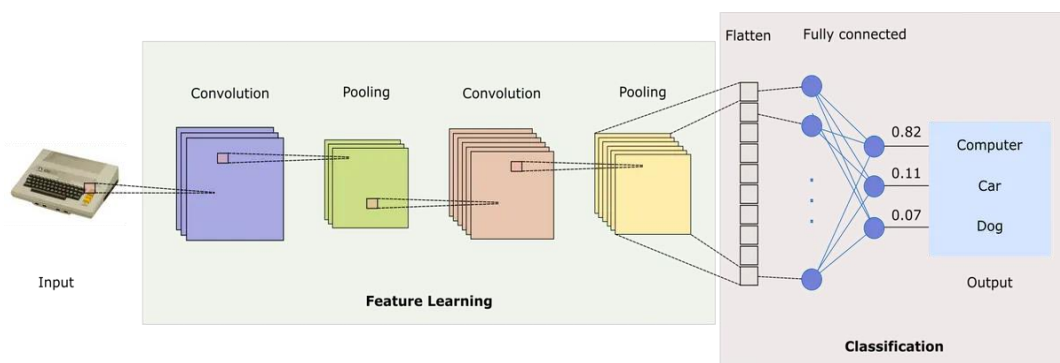
**Tabel 1 Dataset**

	Alutsista	Fasilitas militer	Non-militer	Total
<i>train</i>	488	225	412	1125
<i>validation</i>	97	45	83	225
<i>test</i>	65	30	55	150
Total data	650	300	550	1500



**Gambar 1 Sampel data beserta labelnya**

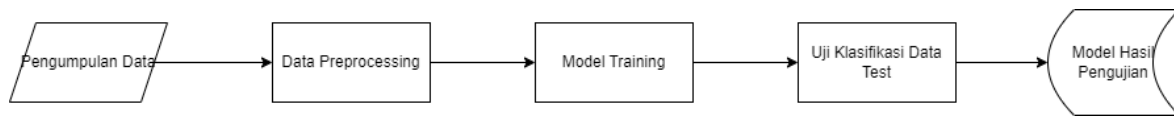
Untuk melakukan deteksi dan sekaligus mengklasifikasikan citra objek melalui pelatihan citra gambar, digunakan algoritma CNN (Convolutional Neural Network). CNN memiliki beberapa layer didalamnya, secara umum digambarkan sebagai berikut.



**Gambar 2 CNN Layers**

Dapat dijelaskan bahwa inputan harus melalui tahap konvolusi di bagian *feature learning*. *Feature* atau fitur merupakan bagian di dalam citra yang diekstrak oleh *filter* untuk kemudian dijadikan *feature map*. Didalam *feature learning*, terdapat *hidden layer*, yang terdiri lapisan-lapisan konvolusi, dan *maxpooling*. Di bagian berikutnya setelah *feature learning*, ada bagian klasifikasi. Bagian klasifikasi melakukan perubahan fitur menjadi 1 dimensi menggunakan *flatten*. Setelah itu, fitur diklasifikasikan melalui sejumlah neuron dan berakhir di neuron output yang jumlahnya sama dengan jumlah kelas yang digunakan untuk mengklasifikasikan. Namun dengan perkembangan teknologi dan penelitian saat ini, arsitektur CNN sudah banyak berkembang. Berbagai jenis arsitektur diciptakan dengan segala kombinasi dan jumlah layer, untuk menghasilkan arsitektur paling baik untuk diterapkan pada model, sehingga model dapat sangat akurat dalam mengklasifikasikan citra gambar.

Dalam penelitian, ada beberapa tahapan yang dilakukan. Berikut merupakan tahapan yang dilalui dalam proses penelitian.



**Gambar 3 Alur Proses Penelitian**

Pada tahap pertama dilakukan pengumpulan data yang terbagi tiga kelas, *deg*, *fac*, dan *non*. Kemudian pada tahap kedua, *data preprocessing*, gambar di *resize* ukurannya secara seragam ke 150 x 150 px agar pada proses input citra di tahap training model, tidak terjadi ketidaksesuaian dimensi gambar. Kemudian pada tahap *model training*, gambar-gambar yang sudah seragam ukurannya dan terbagi menjadi tiga kelas, di pisahkan ke dalam tiga jenis, yakni *data training*, *data validation*, dan *data test*. Kemudian model di *training* dengan *data train* dan *data validation*. Setelah model di-*training* data, *data test* digunakan untuk pengujian model pada tahap uji klasifikasi data test, setelah itu, hasil pengujian menghasilkan model yang telah dibuat.

Untuk arsitektur CNN di atur dengan beberapa layer yang terdiri dari 3 *convolution layers* beserta 3 *down sampling*, beserta 1 *fully connected layer*.

**Tabel 2 Arsitektur CNN**

Layer	Filter	Output
Input	-	150, 150, 3
Conv2D	3 x 3, stride=1, padding	64, 64, 8
ActivationReLU	-	64, 64, 8
MaxPooling2D	2 x 2, strides=2	32, 32, 8
Conv2D	3 x 3, stride=1, padding	32, 32, 16
ActivationReLU	-	32, 32, 16
MaxPooling2D	2 x 2, strides=2	16, 16, 16
Conv2D	3 x 3, stride=1, padding	16, 16, 32
ActivationReLU	-	16, 16, 32
MaxPooling2D	2 x 2, strides=2	8, 8, 32
Dropout	rate=0.5	-
Flatten	-	-
Dense	-	3
Softmax	-	3

Berdasarkan tabel diatas, arsitektur CNN menggunakan input 150 x 150 px, kemudian dilanjutkan proses konvolusi dan *downsampling* menggunakan *MaxPooling2D*, dengan konvolusi menggunakan padding dan stride sebesar 1 px, sedangkan *downsampling* menggunakan stride sebesar 2 dan *pool size* 2x2. Untuk aktivasi menggunakan *ReLU*, setelah melewati hidden layer, citra feature di dropout dengan rate 0.5, serta diubah menjadi 1 dimensi agar tahap klasifikasi dapat dilakukan dengan optimal. Terakhir pada aktivasi output menggunakan softmax dengan ouput sejumlah 3 neuron sesuai jumlah kelas. Fungsi aktivasi *ReLU* (*Rectified Liner unit*) merupakan fungsi non-linier dimana pengaktifan neuron tidak bersamaan, hanya ketika output dari transformasi linier bernilai nol. Fungsi ReLU dituliskan dalam persamaan sebagai berikut.

$$f(x) = \max(0, x)$$

Dimana:

x = nilai data input

f(x) = hasil output fungsi ReLU berupa nilai dalam bentuk 0 dan 1 (Firmansyah & Hayadi Herawan, 2022).

## HASIL DAN PEMBAHASAN

Hasil penelitian berupa model yang telah diuji menggunakan data test, dengan akurasi dan *loss* yang telah terukur. Dengan data citra gambar berupa objek-objek yang memiliki tingkat kemiripan serta letak posisi yang terselipkan diantara rumitnya citra lingkungan sekitar, yang notabene citra gambar tersebut relatif lebih sulit dilatihkan dibanding pada penelitian lain, model diharapkan mampu memiliki tingkat akurasi yang tinggi dan *loss* se-rendah mungkin. Oleh karena itu satu-persatu arsitektur beserta kombinasi *optimizer tool*, diujikan pada data gambar tersebut, untuk memperoleh model yang akurat dan minim kesalahan. Karena sebuah kesalahan akan berujung kepada malapetaka besar yang menimbulkan adanya korban jiwa.

Pada hasil penelitian berikut, model diuji coba menggunakan 5 jenis arsitektur, yakni *Xception*, *InceptionV3*, *VGG16*, *VGG19*, dan *NASNetMobile*. Hasil pelatihan model juga didapatkan berdasarkan berbagai konfigurasi jenis *optimizers*, dan arsitektur, agar didapatkan hasil terbaik dari proses konvolusi yang telah dilakukan. Namun pada tabel berikut ujicoba dilakukan menggunakan *optimizer* yang sama yakni *Adamax*. Berikut ini beberapa percobaan yang dilakukan dengan menggunakan arsitektur yang sudah ada

**Tabel 2 Perbandingan Arsitektur**

Percobaan ke-	Arsitektur	Epoch	Optimizer	Akurasi	Loss
1	Xception	50	Adamax	90,67%	0,3764
2	InceptionV3	50	Adamax	96,00%	0,1757
3	VGG16	50	Adamax	92,67%	0,3142
4	VGG19	50	Adamax	80,67%	0,8647
5	NASNetMobile	50	Adamax	84,67%	0,6550

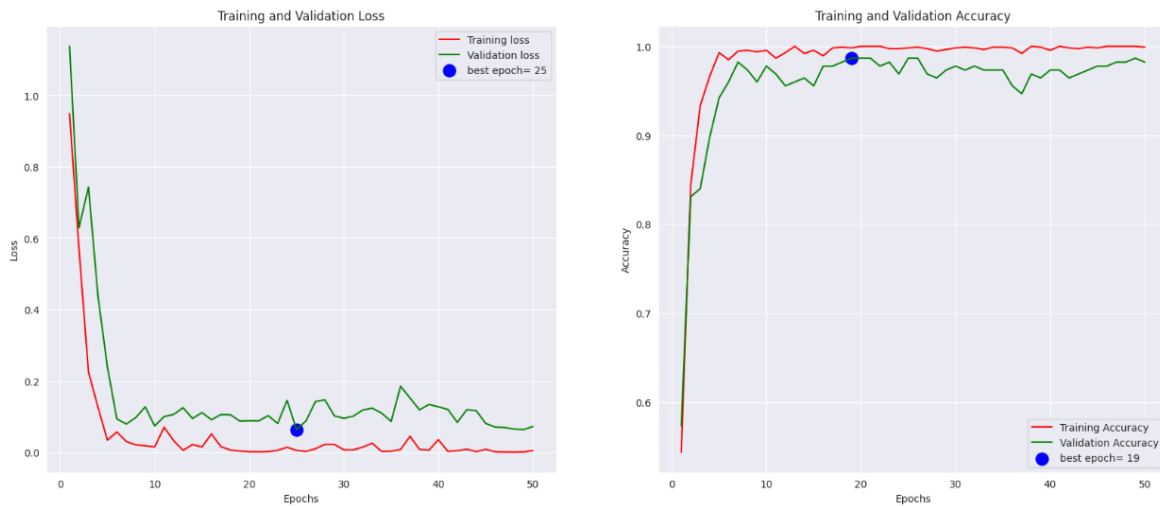
Sumber: TNR 10 Pt

Berdasarkan tabel diatas, terdapat 5 arsitektur yang diuji menggunakan *optimizer Adamax*. Semua training model menerapkan jumlah *epoch* sebanyak 50, hal ini bertujuan untuk menetapkan standar batas pelatihan dari pengujian dari tiap training. Dari kelima arsitektur, *InceptionV3* mampu menunjukkan akurasi yang tertinggi, yakni sebesar 96%, dengan data *loss* terendah sebesar 0,1757. Uji *optimizer* juga dilakukan dengan menggunakan arsitektur berdasarkan hasil terbaik sebelumnya, yakni *InceptionV3*.

**Tabel 3 Optimizers Test**

No	Optimizers	Arsitektur	Epoch	Akurasi	Loss
1	Adadelta	InceptionV3	50	66,00%	0,7720
2	Adadelta	InceptionV3	100	80,00%	0,5066
3	Adam	InceptionV3	50	84,67%	0,6096
4	Adamax	InceptionV3	50	96,00%	0,1757
5	RMSProp	InceptionV3	50	87,33%	0,3375
6	SGD	InceptionV3	50	89,33%	0,3755

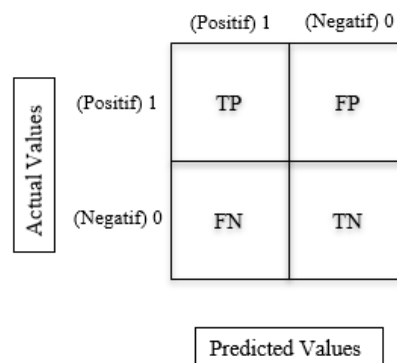
Berdasarkan tabel diatas, *InceptionV3* dengan *optimizer Adamax* tetap memiliki nilai akurasi yang tertinggi yakni 96%, diikuti oleh *SGD* sebesar 89%, *RMSProp* 87%, *Adam* 84%, *Adadelta* 66% dan *Adadelta* (dengan penambahan jumlah *epoch*) 80%. Untuk *Adadelta*, dikarenakan kurva akurasi pada data *train* dan *validation* masih terus naik secara konsisten pada grafik *training*, maka jumlah *epoch* dicoba ditambahkan hingga 100, untuk mengetahui sejauh mana tingkat akurasi yang sesungguhnya. Pada *epoch* 95, kurva mulai cenderung tidak meningkat secara simultan, maka didapat nilai optimum akurasi pada *epoch* 100 sebesar 80%.



**Gambar 4 Grafik training model menggunakan pyplot**

Gambar diatas adalah hasil terbaik pelatihan model menggunakan *InceptionV3* dengan akurasi keseluruhan 96,00% dan *loss* 0,1757. Sebelah kiri merupakan grafik *loss* dari *training* dan *validation* data, sedangkan grafik kanan merupakan akurasi dari *training* dan *validation* data. Sebagai pembandingan, pada penelitian lain yang menggunakan data citra gambar sejenis namun berbeda objek, (Hermawan E, 2023) dengan akurasi 95%, (Listyaputra, t.t.) dengan akurasi 98,38%, (Magdalena, Saidah, Pratiwi, & Putra, 2021) dengan akurasi 95,45%, (Miranda & Aryuni, t.t.) dengan akurasi 98,4%.

Hasil pengujian diperoleh melalui *confusion matrix* dan *classification report*, dimana *confusion matrix* adalah matriks yang menampilkan jumlah data terprediksi, baik yang benar maupun yang salah (Normawati & Prayogi, 2021). *Confusion matrix* berguna untuk menentukan nilai *recall*, *precision*, *specificity*, akurasi yang ditampilkan di *classification report* dan kurva AUC-ROC.



**Gambar 5 Struktur Tabel Confusion Matrix**

Berdasarkan struktur tabel *confusion matrix* diatas, maka *confusion matrix* terdiri dari 4 nilai yakni:

1. TP (*True Positive*) merupakan jumlah prediksi yang benar, dimana model memprediksi kebenaran positif dari suatu data, dan ternyata data tersebut memang positif.
2. TN (*True Negative*) merupakan jumlah prediksi yang benar, dimana model memprediksi kebenaran negatif dari suatu data, dan ternyata data tersebut memang negatif
3. FP (*False Positive*) merupakan jumlah prediksi yang salah, dimana model memprediksi kebenaran positif dari suatu data, namun kenyataannya data tersebut negatif.
4. FN (*False Negative*) merupakan jumlah prediksi yang salah, dimana model memprediksi kebenaran negatif dari suatu data, namun kenyataannya data tersebut positif (Rina, 2023).

Seperti disebutkan sebelumnya, tabel *confusion matrix* berguna untuk menentukan akurasi, *recall*, *precision*, dan spesifisitas. Akurasi adalah nilai yang mengevaluasi seberapa banyak keberhasilan model membuat prediksi yang benar dari semua prediksi yang dilakukan terhadap data *test*. Untuk menghitung nilai akurasi, digunakan rumus sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Sedangkan *precision* atau presisi adalah nilai yang mengevaluasi model, seberapa baik dalam membuat prediksi benar untuk kelas positif dari keseluruhan prediksi positif yang dilakukan. Untuk menghitung nilai presisi, berikut rumusnya.

$$Precision = \frac{TP}{TP + FP}$$

Kemudian *recall* atau sensitivitas merupakan metrik yang mengevaluasi seberapa baiknya model dalam mengidentifikasi kelas positif dengan benar. Untuk menghitung sensitivitas, berikut rumusnya.

$$Recall = \frac{TP}{TP + FN}$$

Selanjutnya adalah spesifisitas. Spesifisitas atau *specificity* merupakan metrik yang menunjukkan seberapa baik model mengklasifikasikan kelas negatif. Untuk menghitung spesifisitas, menggunakan rumus berikut.

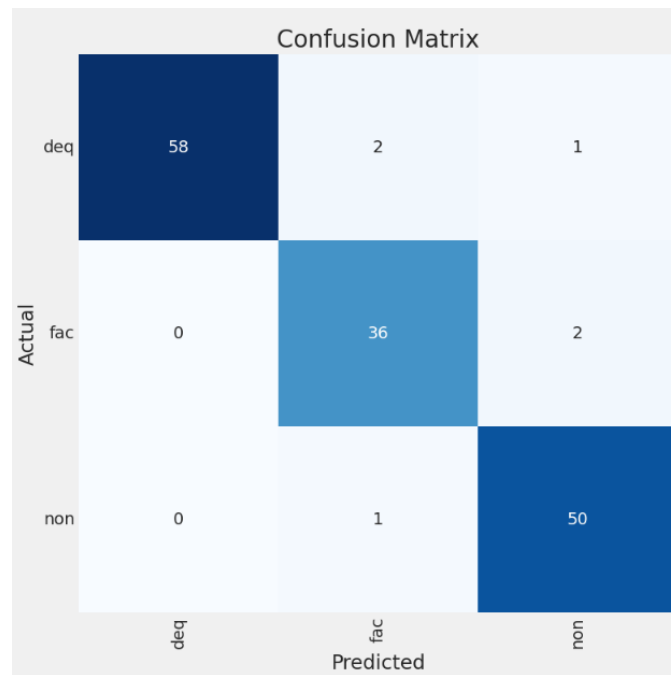
$$Specificity = \frac{TN}{TN + FP}$$

Pada *confusion matrix* 3 kelas seperti dalam kasus penelitian ini, maka struktur tabel *confusion matrix*-nya adalah sebagai berikut.

		deq	fac	non
Actual Values	deq	TP	FP	FP
	fac	FN	TP	TN
	non	FN	TN	TP
		Predicted Values		

**Gambar 6 Struktur Tabel Confusion Matrix 3 Kelas**  
Sumber: (Bharathi, 2023)

Sedangkan *classification report* adalah matriks yang menyajikan evaluasi kinerja *machine learning*. Matriks *classification report* menampilkan nilai *precision*, *recall*, *f1 score*, dan *support* dari pelatihan model. *Precision* merupakan rasio dari total positif benar dan positif salah. *Recall* adalah rasio positif benar dari total positif benar dan negatif salah. Skor f1 merupakan rata-rata harmonik yang ditimbang dari *precision* dan *recall*. Jika nilai f1 semakin mendekati 1.0, maka performa model akan semakin baik. Sedangkan *support* adalah jumlah kemunculan kelas di dalam dataset. Berikut merupakan hasil penelitian dari hasil pelatihan model terbaik dalam *confusion matrix* dan *classification report*.



**Gambar 7 Confusion matrix 3 kelas**

Pada tabel *confusion matrix* diatas, maka terlihat rinciannya sebagai berikut. Model berhasil memprediksi dengan benar kelas *deq* sebanyak 58 kali, memprediksi dengan benar yang bukan termasuk kelas *fac* dengan kenyataannya kelas *deq* sebanyak 2 kali, serta bukan termasuk kelas *non* dengan kenyataan kelas *deq* sebanyak 1 kali. Model tidak melakukan kesalahan prediksi negatif pada kelas *deq*. Model memprediksi dengan benar data bukan kelas *fac* dengan kenyataan data tersebut adalah kelas *non* sebanyak 1 kali, model memprediksi dengan benar bahwa data merupakan kelas *fac* sebanyak 36 kali, model memprediksi dengan benar bahwa data merupakan kelas *non* sebanyak 50 kali, serta model memprediksi dengan benar bahwa data bukan kelas *non* dengan kenyataan data merupakan kelas *fac* sebanyak 2 kali.

```

Classification Report:
-----
              precision    recall  f1-score   support

   deq         1.00        0.95        0.97         61
   fac         0.92        0.95        0.94         38
   non         0.94        0.98        0.96         51

 accuracy                   0.96         150
 macro avg              0.96        0.96        0.96         150
 weighted avg           0.96        0.96        0.96         150
  
```

**Gambar 8 Classification Report**



Pada tabel *classification report*, terlihat untuk nilai *precision* untuk masing-masing kelas *deq*, *fac*, dan *non* adalah 1.00, 0.92, 0.94. Untuk *f1 score*-nya masing-masing adalah 0.97, 0.94, 0.96 artinya nilai terbaik ada pada kelas *deq*, disusul *non*, dan *fac*. Untuk *support* menunjukkan jumlah data *test* masing-masing kelas adalah 61 untuk kelas *deq*, 38 untuk kelas *fac*, dan 51 untuk kelas *non*, sehingga total data *test* adalah 150. Secara *macro* atau keseluruhan, nilai akurasi yang didapat adalah 0.96 atau jika diubah ke persentase, sebesar 96%. Untuk nilai *weighted average* juga relatif sama, yakni sebesar 96%.

## KESIMPULAN

Penelitian penerapan algoritma CNN untuk deteksi dan klasifikasi target militer berdasarkan citra satelit menghasilkan model CNN yang dapat dikatakan memiliki akurasi yang baik. Berdasarkan hasil perancangan dan pengujian model, dengan pemilihan kombinasi terbaik arsitektur InceptionV3 dan optimizer Adamax, didapatkan akurasi model sebesar 96%. Dengan tingkat kehilangan atau loss sebesar 0,17 atau 17%. Dengan hasil seperti itu sudah dikatakan cukup baik. Tetapi jika model akan diterapkan pada penerapan nyata, hasil tersebut tentunya masih sangat kurang baik. Mengingat sasaran yang dideteksi merupakan sasaran yang rumit, serta potensi terjadinya akibat buruk yang sangat berbahaya, maka model masih perlu dikembangkan, serta melalui serangkaian uji keakuratan secara kontinyu. Agar model dapat mencapai tingkat akurasi yang tinggi dan menghindari kesalahan target..

## BIBLIOGRAPHY

- , R. M., Saidah, S., Caecar, K., #3, P., Trisnamulya, A., & #4, P. (t.t.). *JEPIN (Jurnal Edukasi dan Penelitian Informatika) Klasifikasi Tutupan Lahan Melalui Citra Satelit SPOT-6 dengan Metode Convolutional Neural Network (CNN)*.
- Ayu, C., Kinasih, S., & Hidayat, H. (2021a). *Ekstraksi Data Bangunan Dari Data Citra Unmanned Aerial Vehicle Menggunakan Metode Convolutional Neural Networks (CNN) (Studi Kasus: Desa Campurejo, Kabupaten Gresik) Building Data Extraction of Unmanned Aerial Vehicle Image Data Using Convolutional Neural Networks (CNN) Method (Case Study: Campurejo Village, Gresik Regency)*. 17(1), 81–92. Diambil dari <https://iptek.its.ac.id/index.php/geoid/article/view/10289/6517>
- Ayu, C., Kinasih, S., & Hidayat, H. (2021b). *Ekstraksi Data Bangunan Dari Data Citra Unmanned Aerial Vehicle Menggunakan Metode Convolutional Neural Networks (CNN) (Studi Kasus: Desa Campurejo, Kabupaten Gresik) Building Data Extraction of Unmanned Aerial Vehicle Image Data Using Convolutional Neural Networks (CNN) Method (Case Study: Campurejo Village, Gresik Regency)*. 17(1), 81–92.
- Bharathi. (2023, Desember 28). Latest Guide on Confusion Matrix for Multi-Class Classification. Diambil 24 Januari 2024, dari Analytics Vidhya website: <https://www.analyticsvidhya.com/blog/2021/06/confusion-matrix-for-multi-class-classification/>
- Firmansyah, I., & Hayadi Herawan, B. (2022). KOMPARASI FUNGSI AKTIVASI RELU DAN TANH PADA MULTILAYER PERCEPTRON. *JIKO (Jurnal Informatika dan Komputer)*, 6, 200–206. Diambil dari <https://ejournal.akakom.ac.id/index.php/jiko/article/view/600/pdf>
- Hermawan E, A. S. S. D. (2023). 308-318+(1044+Jitter+Dimas+Mulya+Saputra). *JITTER (Jurnal Ilmiah Teknologi Informasi Terapan)*, 9.
- Listyaputra, S. N. (t.t.). *PERBANDINGAN METODE DETEKSI OBJECT BERBASIS CONVOLUTIONAL NEURAL NETWORK PADA CITRA SATELIT LAPAN-A2*.
- Magdalena, R., Saidah, S., Pratiwi, N. K. C., & Putra, A. T. (2021). Klasifikasi Tutupan Lahan Melalui Citra Satelit SPOT-6 dengan Metode Convolutional Neural Network (CNN). *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, 7(3), 335–339.
- Miranda, E., & Aryuni, M. (t.t.). *SISTEMASI: Jurnal Sistem Informasi Klasifikasi Tutupan Lahan Menggunakan Convolutional Neural Network pada Citra Satelit Sentinel-2*. Diambil dari <http://sistemasi.ftik.unisi.ac.id>
- Normawati, D., & Prayogi, S. A. (2021). Implementasi Naïve Bayes Classifier Dan Confusion Matrix

- Pada Analisis Sentimen Berbasis Teks Pada Twitter. Dalam *Jurnal Sains Komputer & Informatika (J-SAKTI)* (Vol. 5). Yogyakarta. Diambil dari <https://ejurnal.tunasbangsa.ac.id/index.php/jsakti/article/view/369/348>
- Rina. (2023, Juni 12). Memahami Confusion Matrix: Accuracy, Precision, Recall, Specificity, dan F1-Score untuk Evaluasi Model Klasifikasi. Diambil 24 Januari 2024, dari <https://esairina.medium.com/memahami-confusion-matrix-accuracy-precision-recall-specificity-dan-f1-score-610d4f0db7cf>
- Yessi Hartiwi, Errissya Rasywir , Yovi Pratama , Pareza Alam Jusia (2020, September 2020) Eksperimen Pengenalan Wajah dengan fitur Indoor Positioning System menggunakan Algoritma CNN. Diambil 25 Januari 2024, dari <http://article.php?article=1812682&val=10513&title=Eksperimen%20Pengenalan%20Wajah%20dengan%20fitur%20Indoor%20Positioning%20System%20menggunakan%20Algoritma%20CNN>
- Sindunuraga Rikarno Putra (2015) Implementation Of Convolutional Neural Network For The Classification Of Object In Images. Diambil 25 Januari 2024, dari <https://core.ac.uk/download/pdf/291471419.pdf>
- M. Ihwanul Iqbal (2022) Deteksi Kerusakan Ban Menggunakan Algoritma Convolutional Neural Network. Diambil 25 Januari 2024, <https://repository.upnjatim.ac.id/9676/37/18081010016.-cover.pdf>
- Arham Rahim, Kusrini, Emha Taufiq Luthfi (2020) Convolutional Neural Network Untuk Kalasifikasi Penggunaan Masker. Diambil 25 Januari 2024, dari [https://www.researchgate.net/profile/KusriniKusrini/publication/348625998\\_Convolutional\\_Neural\\_Network\\_untuk\\_Kalasifikasi\\_Penggunaan\\_Masker/links/6017cbc8299bf1b33e3d5c57/Convolutional-Neural-Network-untuk-Kalasifikasi-Penggunaan-Masker.pdf](https://www.researchgate.net/profile/KusriniKusrini/publication/348625998_Convolutional_Neural_Network_untuk_Kalasifikasi_Penggunaan_Masker/links/6017cbc8299bf1b33e3d5c57/Convolutional-Neural-Network-untuk-Kalasifikasi-Penggunaan-Masker.pdf)
- Sri Wahyuni, Maman Sulaeman (2022, Maret 1) Penerapan Algoritma Deep Learning Untuk Sistem Absensi Kehadiran Deteksi Wajah Di Pt Karya Komponen Presisi. Diambil 25 Januari <https://simantik.panca-sakti.ac.id/index.php/simantik/article/view/127/122>.



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)