

# IMPLEMENTASI ALGORITMA CONVOLUTIONAL NEURAL NETWORK DAN SUPPORT VECTOR MACHINE UNTUK PENDETEKSIAN CANDI JIWA DAN CANDI BLANDONGAN

1<sup>st</sup> Hus'am Badruzzaman  
Universitas Buana Perjuangan  
Karawang, Indonesia  
if19.husambadruzzaman@mhs.ubpkarawang.ac.id

2<sup>nd</sup> Tohirin Al Mudzakir  
Universitas Buana Perjuangan  
Karawang, Indonesia  
tohirin@ubpkarawang.ac.id

3<sup>rd</sup> Rahmat  
Universitas Buana Perjuangan  
Karawang, Indonesia  
rahmat@ubpkarawang.ac.id

**Abstract**— Di wilayah Kabupaten Karawang terdapat peninggalan sejarah yaitu Candi Jiwa dan Candi Blandongan. Keunikan bentuk arsitektur yang dimiliki pada candi ini akan sangat disayangkan apabila tidak diperkenalkan serta diwariskan kepada generasi penerus, dalam rangka melestarikan identitas budaya Indonesia. Dengan memperkenalkan candi melalui pengenalan citra menggunakan algoritma *CNN* dan *SVM*, bertujuan untuk dapat memberikan kontribusi dalam pengembangan teknologi untuk pelestarian dan promosi pariwisata budaya, kepada wisatawan lokal maupun mancanegara. Pada penelitian ini menggunakan 200 data citra Candi Jiwa dan Candi Blandongan yang terbagi kedalam data *training* 80%, data *testing* 10%, dan data validasi 10%. Hasil pengujian yang dilakukan menghasilkan tingkat akurasi sebesar 85%.

**Kata kunci** — Budaya, Candi Jiwa, Candi Blandongan, *CNN*, *SVM*

## I. PENDAHULUAN

Bangsa Indonesia merupakan bangsa yang majemuk serta kaya warisan budaya dan sejarah[1]. Kebudayaan adalah sesuatu yang akan mempengaruhi tingkat pengetahuan dan gagasan yang terdapat dalam pikiran manusia sehingga dalam praktik sehari-hari kebudayaan tersebut bersifat abstrak[2]. Salah satu peninggalan sejarah kebudayaan, yaitu candi. Candi merupakan bahasa yang mengukir gambar dalam bentuk fisik berupa peninggalan zaman dahulu untuk dapat menyampaikan pesan kepada generasi penerus[2]. Kompleks candi batujaya yang berada di Kabupaten Karawang dibangun pada masa Kerajaan Tarumanegara pada abad ke-5 hingga abad ke-6 masehi. Pada situs ini terdapat peninggalan Candi Jiwa dan Candi Blandongan. Secara administratif situs ini terletak di Desa Segaran dan Desa Telagajaya yang dulunya berupa danau, dimana candi dibangun pada tepi danau. Selain itu keunikan lain pada Candi Blandongan yaitu teknologi pembangunan candi dan arsitektur candi yang berbeda dengan candi lainnya yang lebih rumit[3]. Pada Candi Jiwa memiliki bentuk candi yang tidak mempunyai tangga yaitu berupa perbingkai yang rata, pelipit setengah lingkaran, dan perbingkai bergerigi yakni susunan dua lapis bata yang menonjol dan meruncing pada ujungnya[4]. Pentingnya pelestarian candi dan upaya mengenalkannya kepada generasi muda menjadi perhatian penting dalam rangka melestarikan identitas budaya Indonesia.

Penelitian sebelumnya terkait pengenalan jenis candi berdasarkan bentuk dan modelnya menggunakan metode *CNN* ada Yollo v3. Pada metode ini *dataset* yang telah dikumpulkan diproses ke tahap *labeling* kemudian diberikan anotasi pada setiap gambar candi untuk mendapatkan koordinat objek yang akan dilatih. Setelah itu akan masuk ke proses *training* dengan presentase 80% data *training* dan 20% data *testing* menggunakan *framework darknet* dengan menyimpan *weight* setiap 1000 iterasi. Pada iterasi 6000 *batch* menghasilkan *avg loss* sebesar 0,0500 dan pengujian menggunakan 10 gambar candi menghasilkan tingkat akurasi sebesar 100%[5]. Serta penelitian lainnya menggunakan algoritma *Mask R-CNN* untuk mengenali objek citra situs cagar budaya Candi Mahligai. Pada metode ini menggunakan *dataset* sebanyak 150 data citra yang terdiri dari 2 kelas yaitu Candi Mahligai dan bukan Candi Mahligai. *Dataset* yang digunakan menggunakan gambar citra candi yang terbagi menjadi 100 data *training* dan 50 data *testing*. Pada penelitian yang dilakukan dengan mengambil satu kelas *dataset* untuk nantinya diidentifikasi sebagai Candi Mahligai. Hasil pengujian yang dilakukan menggunakan 50 data citra menghasilkan nilai *loss* 0,18 pada *epoch* ke 1000 dengan tingkat akurasi sebesar 82%[6]. Selain itu pada penelitian terkait sistem klasifikasi relief Candi Borobudur menggunakan *Local Binary Pattern* dan Algoritma *Nearest Neighbors*. *Dataset* pada penelitian yang dilakukan menggunakan data latih sebanyak 50 data yang terbagi kedalam 50 kelas, dan data uji sebanyak 300 data yang diambil dari 6 data setiap kelas. *Dataset* yang sudah melalui proses *cropping* kemudian dilakukan ekstraksi ciri menggunakan *Local Binary Pattern* kemudian diklasifikasikan menggunakan metode *Nearest Neighbors* menghasilkan tingkat akurasi sebesar 66,33%[7]. Pada penelitian perancangan sistem visualisasi Candi Tawang Alun berbasis *Augmented Reality* sistem akan menampilkan objek 3D Candi Tawang Alun[8]. Kemudian dalam studi penelitian komparatif model 3 dimensi menggunakan teknologi *terrestrial laser scanner (TLS)* *Leica RTC360* dan *Leica BLK2GO* menggunakan metode *close range photogrammetry* dan *terrestrial laser scanner*, dengan uji statistik *paired sample test*, hasil model 3 dimensi dengan menggunakan teknologi *Leica RTC360* dan *Leica BLK2GO* memiliki perbedaan rata-rata sampel jarak antara kedua teknologi. Berdasarkan perbandingan teknologi *Leica RTC360* dengan teknologi *Leica BLK2GO* dari segi ketelitian geometri dan visualisasi model 3 dimensi yang dibentuk, teknologi *Leica RTC360* lebih baik daripada *Leica BLK2GO* untuk kebutuhan rekonstruksi pemodelan 3 dimensi Candi Mendut[9].

Berdasarkan permasalahan pendeteksian objek candi pada penelitian sebelumnya, penelitian ini bertujuan untuk dapat memberikan kontribusi dalam pengembangan teknologi untuk pelestarian dan promosi pariwisata budaya Candi Jiwa dan Candi Blandongan. Berbeda dengan penelitian sebelumnya pada penelitian ini menggunakan fitur ekstraksi yang didapat dari model *convolutional neural network* yang akan digunakan sebagai *input* pada model *support vector machine* untuk deteksi jenis candi.

## II. DATA DAN METODE

### A. Candi

Candi merupakan bangunan peninggalan peradaban Hindu Budha sebagai tempat keagamaan, gapura, keraton, tempat pemandian maupun sebagai tempat peribadahan[2]. Candi adalah suatu kebudayaan yang bersifat bendawi yang memuat nilai-nilai kebudayaan penting untuk disampaikan[10]. Candi jiwa merupakan salah satu bangunan bersejarah yang berada di area kompleks percandian batujaya. candi ini memiliki bentuk persegi empat bila dilihat sisi atasnya yang berbentuk menyerupai bunga teratai, candi jiwa ini juga sering disebut unur jiwa. Pada candi ini tidak ditemukan undakan sehingga bentuknya menyerupai stupa atau arca Budha di atas bunga teratai yang sedang mekar dan terapung di atas air[4].

Kemudian ada Candi Blandongan. Candi ini memiliki bentuk bujur sangkar berukuran 24,2 meter x 24,2 meter. Berdasarkan kepada penelitian terhadap Candi Blandongan dapat diketahui bahwa Candi Blandongan adalah candi utama dari kompleks candi. Berdasarkan ukuran candi, keberadaan pintu masuk pada keempat sisi candi, masing-masing sisi tetap sunyi di sudut tenggara, barat daya, timur laut dan barat laut dari titik mata angin. Pintu-pintu itu mengarah ke tengah ke kuil untuk melakukan upacara keagamaan atau membuat pengorbanan[3].

### B. Pengolahan Citra Digital

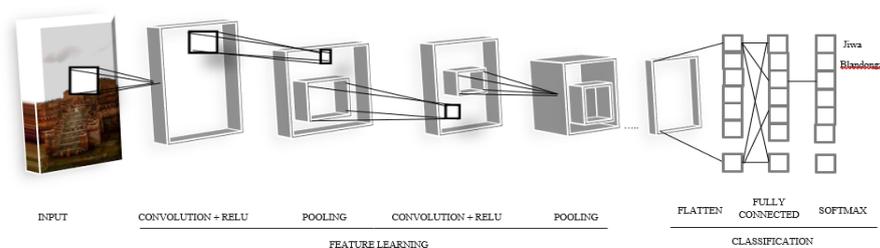
Citra adalah gambar dua dimensi yang dihasilkan dari gambar analog dua dimensi yang kontinu menjadi gambar diskrit melalui proses *sampling*. Proses *sampling* dibagi menjadi 2 yaitu *downsampling* dan *upsampling*. *Downsampling* merupakan proses untuk menurunkan jumlah piksel atau resolusi citra sedangkan *upsampling* merupakan proses untuk menaikkan jumlah piksel atau peningkatan resolusi citra [11].

### C. Klasifikasi

Pengklasifikasian gambar merupakan tugas yang sangat mudah bagi manusia, akan tetapi bagi komputer merupakan tugas yang sangat kompleks sehingga klasifikasi objek/gambar menjadi permasalahan utama dalam bidang *computer vision*[12]. Klasifikasi yaitu suatu karakteristik untuk sebuah proses pengelompokan atau pembagian pada data yang sama ke dalam beberapa kelas[13].

### D. Convolutional Neural Network

*Convolutional Neural Network (CNN)* adalah salah satu pengembangan dari jaringan saraf tiruan yang terinspirasi dari jaringan saraf manusia serta sering digunakan dalam data gambar untuk mendeteksi dan mengenali objek. Berikut ini merupakan alur proses *CNN* [14].

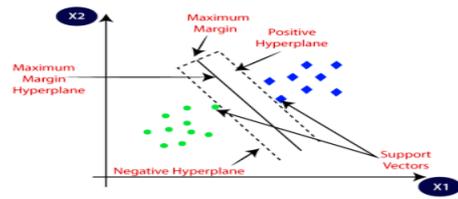


Gambar 2. 1 Arsitektur CNN  
Sumber : (Trivusi, 2022)

Lapisan ekstraksi fitur citra terletak pada lini arsitektur yang tersusun atas beberapa susunan serta tiap susunan tersusun dari *neuron* yang terkoneksi pada wilayah lokal lapisan sebelumnya. Lapisan awal merupakan *convolution layer* serta lapisan kedua merupakan *pooling layer*. Pada keluaran tiap lapisan digunakan suatu peranan aktivasi. *Fully connected layer* mencampurkan proses dari *Convolution*, *ReLU*, *Pooling*, serta *Flatten*. Hasil dari *flatten* dimasukkan satu persatu[15].

### E. Support Vector Machine

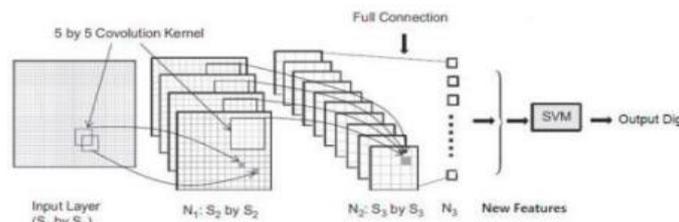
*Support Vector Machine (SVM)* merupakan algoritma klasifikasi untuk data *linier* dan *non linier*. Algoritma *SVM* merupakan algoritma yang bekerja menggunakan pemetaan *non linier* untuk merubah data pelatihan asli menjadi data pelatihan dengan dimensi yang lebih tinggi. Dalam perubahan dimensi baru ini akan mencari *hyperplane* untuk memisahkan optik linier[16].*SVM* memberikan model klasifikasi yang solusinya global optimal, yaitu selalu memberikan model yang sama dan solusi dengan *margin* maksimal[17].



Gambar 2. 2 *Hyperplane SVM*  
Sumber : (Trivusi, 2022)

F. Model *Hybrid CNN* dan *SVM*

Model *Hybrid CNN* dan *SVM* adalah model yang menggabungkan dua algoritma *machine learning* yaitu *Convolutional Neural Network (CNN)* dan *Support Vector Machine (SVM)*. *CNN* merupakan algoritma yang dapat mempelajari fitur lokal *invariant* dengan sangat baik dan dapat mengekstrak informasi paling diskriminatif dari sebuah gambar. Sementara pengklasifikasian *SVM* memiliki kemampuan yang baik dalam meminimalkan kesalahan generalisasi pada data yang tersembunyi serta algoritma yang baik untuk kasus klasifikasi *biner*. Model *hybrid CNN* dan *SVM* diusulkan sebagai pengganti lapisan *softmax* pada *CNN*. *CNN* berfungsi sebagai ekstraksi fitur serta *SVM* sebagai pengklasifikasian *biner* [18]. Arsitektur *hybrid* model *CNN* dan *SVM* sebagai berikut.



Gambar 2. 3 Arsitektur Model *Hybrid CNN* dan *SVM*  
Sumber : (Ahlawat dan Choudhary, 2020)

G. *Confusion Matrix*

*Confusion matrix* adalah sebuah tabel termasuk jumlah informasi pengujian yang diklasifikasikan dengan benar dan salah [19]. *Confusion matrix* adalah alat visualisasi yang dapat digunakan untuk penilaian kemampuan model klasifikasi untuk melakukan klasifikasi evaluasi komparatif, sehingga digunakan *confusion matrix* dalam penelitian ini. Matrik sulit mencakup informasi tentang kategori aktual dan kategori yang diprediksi. Informasi ini akan didapatkan di tabel matrik yang menggantikan kategori yang diperkirakan, dan setiap rangkaian mewakili peristiwa dalam kategori itu. *Confusion matrix* merupakan alat ukur yang berbentuk matriks untuk mendapatkan akurasi dari algoritma yang digunakan[20].

Tabel 2. 1 *Confusion Matrix*

<i>Confusion Matrix</i>	<i>Predicted Class</i>	
	<i>Positive</i>	<i>Negative</i>
<i>Actual Positive</i>	TP	FN
<i>Actual Negative</i>	FP	TN

Keterangan :

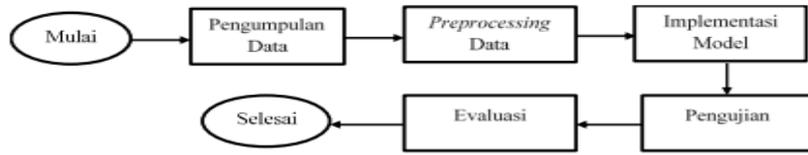
- (TP) *True positive* : Apabila data yang diperkirakan bernilai positif dan nilai sebenarnya positif.
- (FP) *False Positive* : Apabila apa yang dikatakan tidak seperti dengan nilai sebenarnya.
- (FN) *False Negative* : Apabila data yang diperkirakan bernilai negatif dan sebenarnya positif.
- (TN) *True Negative* : Apabila terbukti antara yang diperkirakan negatif dan sebenarnya negatif.

Berikut merupakan rumus menentukan nilai akurasi pada model *CNN* dan *SVM*.

$$Akurasi = \frac{Jumlah\ prediksi\ benar}{Jumlah\ total\ prediksi} \times 100\% \quad (1)$$

H. Prosedur Penelitian

Berikut ini merupakan uraian tahapan prosedur penelitian yang dilakukan.



Gambar 2. 4 Prosedur Penelitian

III. HASIL DAN PEMBAHASAN

A. Pengumpulan Data

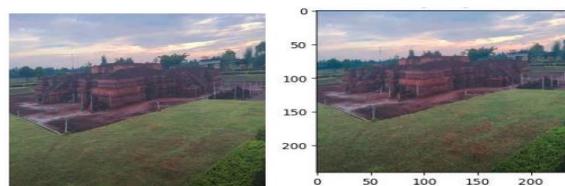
Pengumpulan data pada penelitian ini diperoleh dari situs *google image* menggunakan metode *scrapping* data berupa gambar candi di Karawang. Jumlah data yang digunakan menggunakan 200 data citra candi yang terdiri dari Candi Jiwa dan Candi Blandongan.



Gambar 3. 1 (a) Sampel data Candi Jiwa (b) Sampel data Candi Blandongan

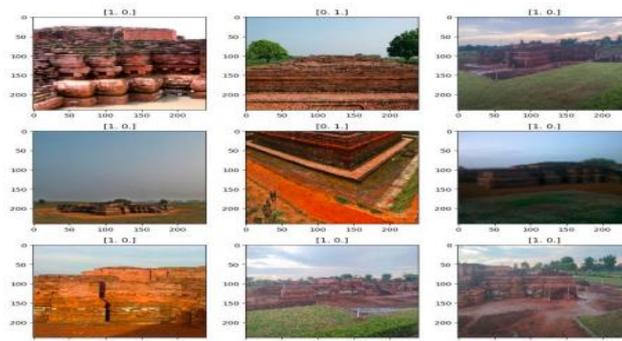
B. Preprocessing Data

Pada tahap *preprocessing* data yang pertama yaitu *resize* ukuran gambar menjadi ukuran 240x240 piksel. Ukuran gambar yang seragam dapat memudahkan didalam pengolahan data.



Gambar 3. 2 Hasil *Resize*

Kemudian setelah *resize* gambar selanjutnya yaitu mengkonversi data dan label gambar menjadi bentuk *array* menggunakan *numpy array*, berfungsi untuk mempercepat proses komputasi serta manipulasi data. Setelah itu *encoding* label yaitu merubah label kategori menjadi representasi *numerik* menggunakan *Label Binarizer*. Kemudian menggunakan fungsi *shuffle* untuk mengacak urutan data dan label, untuk mencegah bias, meminimalkan efek data terurut serta meminimalkan resiko *overfitting*.



Gambar 3. 3 Hasil *Encoding* Label candi

Proses selanjutnya didalam tahap *preprocessing* yaitu membagi data citra candi kedalam 3 kategori yaitu data *training* sebanyak 80%, data *testing* sebanyak 10% dan data validasi sebanyak 10%. Bertujuan untuk menghasilkan model yang dapat melakukan generalisasi pada data yang belum pernah dilihat sebelumnya pada proses *training* dan *testing*.

Tabel 3. 1 Hasil Skenario Pembagian Dataset candi

Skenario Pembagian Dataset		
	Perbandingan	Jumlah Data
Data <i>Training</i>	80%	160
Data <i>Testing</i>	10%	20
Data <i>Validasi</i>	10%	20

Setelah *dataset* citra candi dibagi kedalam masing-masing *subset* kategori, langkah selanjutnya dengan menormalisasi data, berfungsi untuk memastikan setiap elemen data berada pada dalam rentang nilai 0 dan 1. Sehingga model akan dapat mempelajari data citra dengan lebih efisien dan hasil yang lebih baik.

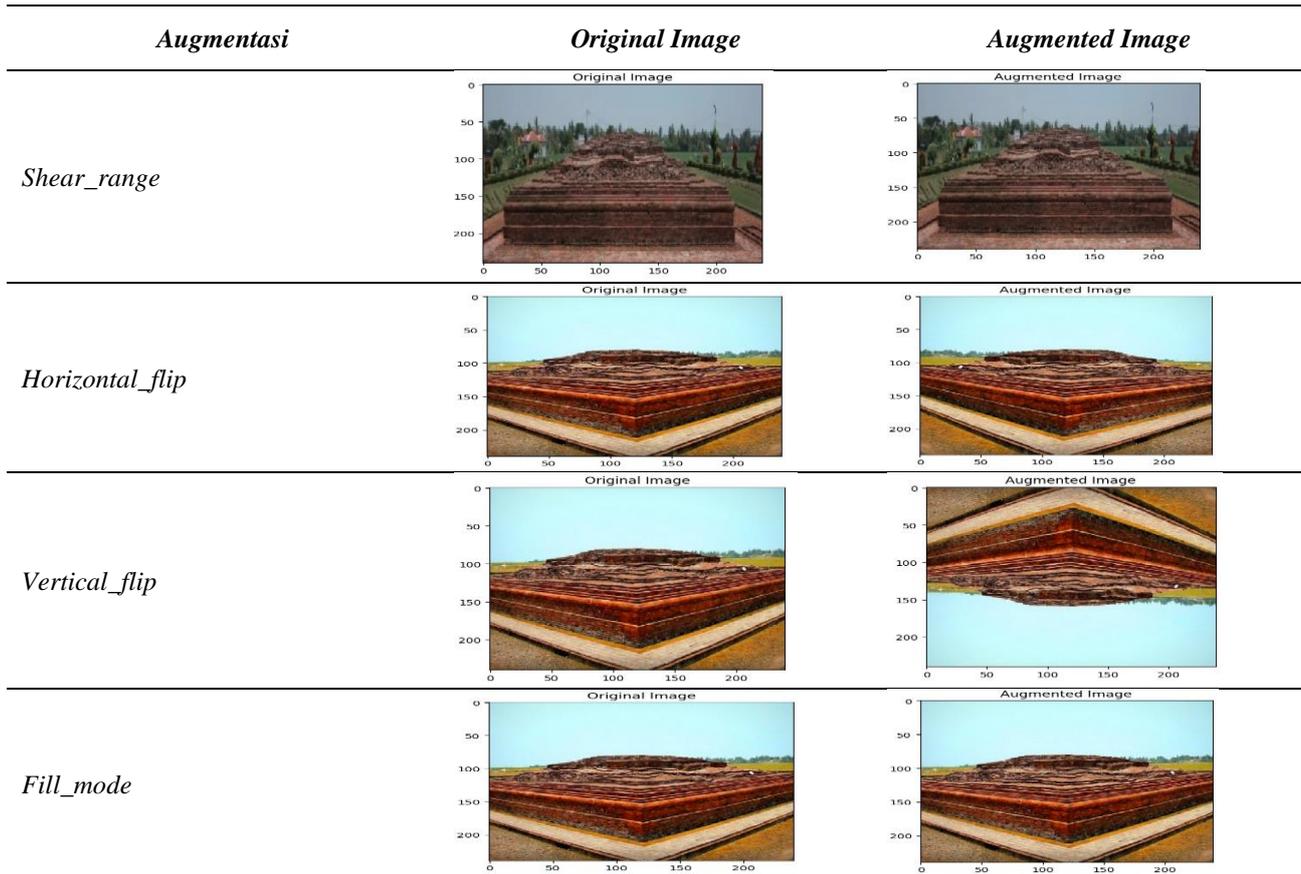
```
Shape data training: (160, 240, 240, 3)
Shape data testing: (20, 240, 240, 3)
Shape data validasi: (20, 240, 240, 3)
```

Gambar 3. 4 Hasil normalisasi dataset candi

Tahap *preprocessing* data selanjutnya yaitu menerapkan *augmentasi* data untuk mencegah *overfitting*, serta menambah jumlah data dengan memodifikasi citra menggunakan *Image Data Generator*. Hasil *augmentasi* pada tabel berikut.

Gambar 3. 5 Hasil *Augmentasi* Data

<i>Augmentasi</i>	<i>Original Image</i>	<i>Augmented Image</i>
<i>Rotation_range</i>		
<i>Zoom_range</i>		
<i>Width_shift_range</i>		
<i>Height_shift_range</i>		



Hasil dari tahap *augmentasi* citra dilakukan dengan tahapan seperti *rotation\_range* untuk merotasi gambar dengan rentang  $20^\circ$ , *zoom\_range* untuk memperbesar ukuran gambar sebesar 20%, *width\_shift\_range* untuk melebarkan ukuran gambar sebesar 20%, *height\_shift\_range* untuk mengatur tinggi gambar sebesar 20%, *shear\_range* untuk mengatur pergeseran sumbu gambar 15%, *horizontal\_flip* perputaran gambar secara horisontal, *vertical\_flip* perputaran gambar secara vertikal, *fill\_mode* untuk mengisi bagian terbuka pada gambar saat dilakukan transformasi gambar.

### C. Implementasi Model

Pada tahap ini dilakukan pembuatan model *hybrid CNN* dan *SVM* untuk melakukan klasifikasi Candi Jiwa dan Candi Blandongan. Pembuatan model menggunakan perangkat lunak *google collab*. *CNN* akan digunakan sebagai pengekstraksi fitur dari gambar dan *SVM* akan digunakan untuk *output* klasifikasi di akhir. Arsitektur yang pada penelitian ini meliputi 2 tahap yaitu pembuatan model *CNN* dan *SVM* sebagai berikut:

#### 1) Tahap Pembuatan model *CNN*

*Convolutional* pertama, pada layer ini operasi menggunakan input gambar ukuran 240x240 piksel. Menggunakan inputan 16 filter dan *kernel* 6x6. Setelah itu pada proses konvolusi ditambahkan sebuah aktivasi *ReLU* untuk menghilangkan nilai negatif. *Pooling layer* pertama, setelah proses konvolusi selanjutnya masuk ke *pooling layer* menggunakan *max pooling* dengan filter 2x2 untuk mengambil nilai maksimum dalam setiap piksel. Hal ini membantu untuk mengurangi jumlah parameter dan komputasi yang diperlukan. *Convolutional* kedua, pada proses konvolusi kedua menggunakan 32 filter dengan *kernel* 6x6 untuk menghasilkan fitur-fitur yang lebih kompleks dari gambar serta menggunakan fungsi aktivasi *ReLU* untuk memperkenalkan *non-linearitas* yang memungkinkan model untuk mempelajari hubungan yang lebih kompleks antara fitur-fitur *input* dan *output* yang diinginkan. *Pooling layer* kedua, menggunakan *max pooling* dengan filter 2x2 untuk mengambil nilai maksimum dalam setiap piksel. *Convolutional* ketiga, pada proses konvolusi kedua menggunakan 64 filter dengan *kernel* 6x6 untuk menghasilkan fitur-fitur yang lebih kompleks dari gambar. *Pooling layer* ketiga, menggunakan *max pooling* dengan filter 2x2 untuk mengambil nilai maksimum dalam setiap piksel. *Convolutional* keempat, pada proses konvolusi kedua menggunakan 128 filter dengan *kernel* 6x6 untuk menghasilkan fitur-fitur yang lebih kompleks dari gambar. *Pooling layer* keempat, menggunakan *max pooling* dengan filter 2x2 untuk mengambil nilai maksimum dalam setiap piksel. *Convolutional* kelima, pada proses konvolusi kedua menggunakan 256 filter dengan *kernel* 6x6 untuk menghasilkan fitur-fitur yang lebih kompleks dari gambar. *Pooling layer* kelima, menggunakan *max pooling* dengan filter 2x2 untuk mengambil nilai maksimum dalam setiap piksel. *BatchNormalization*, pada proses ini dilakukan normalisasi pada setiap *batch* data yang melewati model. Hal ini untuk membantu meningkatkan stabilitas dan kecepatan pelatihan model. *Flatten*, pada proses ini terjadi perubahan *tensor input* menjadi *single vector*. Ini diperlukan sebelum menghubungkan *output* dari *layer* konvolusi ke *dense layer*. *Dropout*, selanjutnya dilakukan *dropout* dengan nilai 0,2 atau 20% yang bertujuan untuk memilih acak beberapa neuron yang tidak digunakan selama proses pelatihan serta mencegah *overfitting* pada saat proses *training*. *Dense* pertama, pada *layer dense* ini dengan 1024 *neuron* dan penambahan aktivasi *ReLU* bertujuan untuk mempelajari representasi fitur yang lebih kompleks dari fitur-fitur yang telah diekstraksi sebelumnya. *Dense*

terakhir, pada *layer dense* terakhir menggunakan 2 *neuron* dengan fungsi aktivasi *sigmoid*. Untuk menghasilkan *output* dalam bentuk probabilitas untuk kedua kelas yang ada. Setelah pembuatan model *CNN* dilakukan akan menghasilkan beberapa parameter sebagai berikut.

Tabel 3. 2 Hasil parameter model *CNN*

<i>Layer Type</i>	<i>Output Shape</i>	<i>Param #</i>	<i>Keterangan</i>
Conv2D	None,235,235,16	1744	Layer ke-1
Maxpooling2D	None,117,117,16	0	Layer ke-2
Conv2D	None,112,112,32	18464	Layer ke-3
Maxpooling2D	None,56,56,32	0	Layer ke-4
Conv2D	None,51,51,64	73792	Layer ke-5
Maxpooling2D	None,25,25,64	0	Layer ke-6
Conv2D	None,20,20,128	295040	Layer ke-7
Maxpooling2D	None,10,10,128	0	Layer ke-8
Conv2D	None,5,5,256	1179904	Layer ke-9
Maxpooling2D	None,2,2,256	0	Layer ke-10
BatchNormalization	None,2,2,256	1024	Layer ke-11
Flatten	None,1024	0	Layer ke-12
Dropout	None,1024	0	Layer ke-13
Dense	None,1024	1049600	Layer ke-14
Dense	None,2	2050	Layer ke-15
<b>Total Params</b>	2.621.618		
<b>Trainable Params</b>	2.621.106		
<b>Non-trainable params</b>	512		

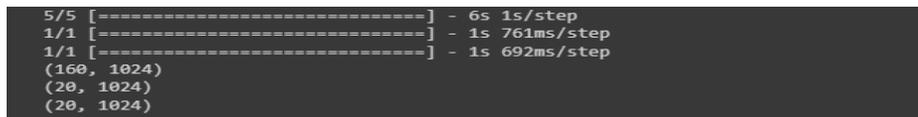
2) Tahap pembuatan model *SVM*

Model *SVM* dengan *kernel liniear* dilatih menggunakan fitur yang diekstraksi dari data *training*. Kemudian akan menghasilkan klasifikasi gambar Candi Jiwa dan Candi Blandongan. Tahapan pelatihan model *SVM* sebagai berikut.

*ExtractCNN*, untuk membuat model baru yang akan mengambil *input* dari model *CNN* dan menghasilkan *output* dari *layer* sebelum *dense layer*. Hal ini dilakukan untuk mendapatkan fitur-fitur ekstraksi yang dihasilkan oleh model *CNN*. *Feat\_trainCNN*, menggunakan model dari *extractCNN* fitur ekstraksi dari data *training* diekstraksi kemudian disimpan dalam variabel *feat\_trainCNN*. *Feat\_testCNN*, fitur ekstraksi dari data *testing* juga akan diekstraksi menggunakan model *extractCNN* dan disimpan kedalam variabel *feat\_testCNN*. *Feat\_validCNN*, fitur ekstraksi dari data validasi diekstraksi menggunakan *extractCNN* dan disimpan kedalam variabel *feat\_validCNN*.

```
extractCNN = Model(inputs=cnn_model.inputs, outputs=cnn_model.layers
[-4].output)
feat_trainCNN = extractCNN.predict(x_train)
feat_testCNN = extractCNN.predict(x_test)
feat_validCNN = extractCNN.predict(x_val)
print(feat_trainCNN.shape)
print(feat_testCNN.shape)
print(feat_validCNN.shape)
```

Dengan menggunakan *filter* 6x6 pada *layer* konvolusi pertama model *CNN* data gambar yang digunakan dalam *training* dan *testing* telah diubah menjadi vektor dengan panjang 1024. Terdapat 5 *batch* data citra secara keseluruhan, dengan waktu *training* per *batch* membutuhkan waktu 1 detik.



Gambar 3. 6 Hasil Ekstraksi fitur *CNN*

Selanjutnya setelah mendapatkan fitur ekstraksi *CNN* kemudian akan digunakan sebagai *input* untuk *training* model *SVM* untuk klasifikasi citra candi. Model *SVM* yang dibangun menggunakan *kernel liniear*, *kernel liniear* digunakan untuk menghasilkan fungsi pemetaan *liniear* yang sederhana antara fitur *input* dengan ruang karakteristik yang diperluas. Sehingga akan mendapatkan *hyperplane* yang maksimal antar kelas data.

D. Pengujian

Pengujian dilakukan pada data *training* dan *validation*, pengujian menggunakan *hyperparameter* yang telah ditentukan sebelumnya yaitu jumlah *batch\_size* sebesar 32, jumlah *epoch* 100, jumlah *patience* 5, dan *monitoring val\_loss*, serta menggunakan fungsi *early stopping* untuk mencegah terjadinya *overfitting/underfitting*. Fungsi *early stopping* pada pengujian model klasifikasi diperlukan agar nantinya saat proses pengujian model apabila tidak ada perubahan dalam batasan yang telah

ditentukan maka proses *training* model akan dihentikan. Selain itu pada fungsi *early stopping* menggunakan parameter *restore\_best\_weight* yaitu model akan mengambil dan menyimpan bobot terbaik selama pelatihan, serta mengembalikan bobot terbaik

pada saat pelatihan untuk memprediksi data baru.

```
Epoch 6/100
5/5 [=====] - 29s 6s/step - loss: 0.5253 - accuracy: 0.7188 - val_loss: 0.6850 - val_accuracy: 0.6000
Epoch 7/100
5/5 [=====] - 30s 6s/step - loss: 0.5555 - accuracy: 0.7063 - val_loss: 0.6899 - val_accuracy: 0.6500
Epoch 8/100
5/5 [=====] - 29s 6s/step - loss: 0.4751 - accuracy: 0.8000 - val_loss: 0.6872 - val_accuracy: 0.5500
Epoch 9/100
5/5 [=====] - 31s 6s/step - loss: 0.5083 - accuracy: 0.7688 - val_loss: 0.6819 - val_accuracy: 0.5500
Epoch 10/100
5/5 [=====] - 29s 6s/step - loss: 0.4899 - accuracy: 0.7750 - val_loss: 0.6852 - val_accuracy: 0.6500
Epoch 11/100
5/5 [=====] - 29s 6s/step - loss: 0.5342 - accuracy: 0.7188 - val_loss: 0.6859 - val_accuracy: 0.6500
Epoch 12/100
5/5 [=====] - 30s 6s/step - loss: 0.4876 - accuracy: 0.7437 - val_loss: 0.6881 - val_accuracy: 0.6500
Epoch 13/100
5/5 [=====] - 29s 5s/step - loss: 0.4579 - accuracy: 0.7812 - val_loss: 0.6865 - val_accuracy: 0.6000
Epoch 14/100
5/5 [=====] - 30s 6s/step - loss: 0.4137 - accuracy: 0.8188 - val_loss: 0.6825 - val_accuracy: 0.7000
```

Gambar 3. 7 Hasil perhitungan *epoch*

Pada gambar 3.7 fungsi *early stopping* dilakukan pada *epoch* ke-14, pada *epoch* ke-14 dihentikan proses pelatihannya karena sudah mencapai batas yang ditentukan. Sehingga model tidak perlu melakukan pelatihan sampai *epoch* terakhir / *epoch* ke-100, karena pada *epoch* ke-14 sudah mendapatkan model terbaik. Pengujian model *CNN* mendapatkan tingkat akurasi sebesar 81%, akurasi validasi sebesar 70%. Serta nilai *loss training* 41% dan *loss validasi* 68%.

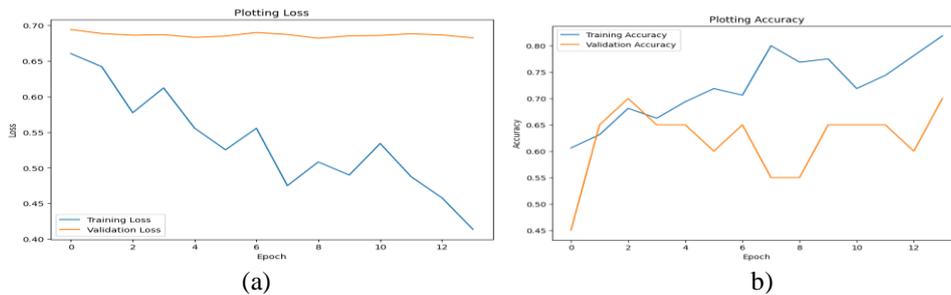
```
1/1 [=====] - 1s 895ms/step - loss: 0.6644 - accuracy: 0.6000
Loss pada data pengujian: 0.6643850207328796
Akurasi pada data pengujian: 0.6000000238418579
```

Gambar 3. 8 Hasil evaluasi *loss* pengujian model *CNN*

```
1/1 [=====] - 1s 1s/step - loss: 0.6819 - accuracy: 0.5500
Loss pada data validasi: 0.6819105744361877
Akurasi pada data validasi: 0.550000011920929
```

Gambar 3. 9 Hasil validasi akurasi pengujian model *CNN*

Berdasarkan evaluasi dari pegujian model *CNN* gambar 3.8 dan gambar 3.9 pada data citra Candi Jiwa dan Candi Blandongan menghasilkan tingkat *loss* sebesar 66% dan akurasi sebesar 60% serta validasi *loss* 68% dan validasi akurasi 55%.



Gambar 3. 10 (a) Grafik *loss* (b) Grafik akurasi

Berdasarkan hasil pengujian model *loss* dan akurasi *CNN* dapat dilihat pada grafik gambar 4.13 berikut, grafik tingkat akurasi dan *loss* pada model *CNN* yang dihasilkan cenderung fluktuatif serta kurang maksimal, namun grafik menunjukkan hasil yang baik dimana akurasi pada data *training* maupun data validasi tidak menunjukkan adanya gejala *overfitting* , yaitu akurasi pada data training akan terus meningkat sedangkan akurasi pada data validasi akan stagnan. Namun hal ini tidak terjadi pada proses pelatihan model pada penelitian ini. Hal yang sama juga terjadi pada grafik *loss* tidak ada gejala *overfitting*. Walaupun terlihat fluktuatif namun grafik menunjukkan bahwa model meminimalkan tingkat *loss*. Jika dilihat dari grafik akurasi dan *loss* dapat dikatakan bahwa jumlah iterasi 100 merupakan jumlah yang tepat untuk pelatihan model *CNN* pada penelitian ini. Selain itu *hyperparameter* yang ditentukan sebelumnya dapat bekerja dengan baik sehingga mendapatkan hasil yang baik.

```
SVM Training Accuracy Score: 83.125
SVM Testing Accuracy Score: 85.0
SVM Validation Accuracy Score: 75.0
```

Gambar 3. 11 Hasil pengujian akurasi model *SVM*

Dari hasil pengujian akurasi *SVM* pada data *training* menunjukkan nilai 83%, data *testing* 85% dan data validasi 75%.

Berdasarkan pengujian yang telah dilakukan pada model *CNN* dan model *SVM* selanjutnya yaitu dengan menggabungkan kedua hasil pengujian model pada *CNN* dan *SVM* sebagai berikut.

```
print("--Training Accuracy..")
print("CNN-SVM Training Accuracy:
{:.2f}%".format(TrainSVMScoreCNN))
print("\n--Testing Accuracy..")
print("CNN-SVM Testing Accuracy:
{:.2f}%".format(TestSVMScoreCNN))
print("\n--Validation Accuracy..")
print("CNN-SVM Validation Accuracy:
{:.2f}%".format(ValidSVMScoreCNN))
```

Berdasarkan pengujian yang dilakukan model *CNN* dan *SVM* berhasil dibuat hasil akhir yang didapatkan yaitu akurasi *training* sebesar 83%, *testing* 85%, dan validasi 75%.

```
--Training Accuracy..
CNN-SVM Training Accuracy: 83.12%

--Testing Accuracy..
CNN-SVM Testing Accuracy: 85.00%

--Validation Accuracy..
CNN-SVM Validation Accuracy: 75.00%
```

Gambar 3. 12 Hasil akurasi model *CNN* dan *SVM*

#### E. Evaluasi

Berdasarkan hasil pengujian yang dilakukan terhadap gambar Candi Jiwa dan Candi Blandongan sebanyak 20 gambar yang diprediksi. *True Positif (TP)* merupakan gambar Candi Jiwa, dan model mengklasifikasikan Candi Jiwa. Kemudian *True Negative (TN)* merupakan gambar Candi Blandongan, dan model mengklasifikasikan Candi Blandongan. Setelah itu *False Positive (FP)* merupakan gambar Candi Blandongan, dan model mengklasifikasikan Candi Jiwa. Dan yang terakhir *False Negative (FN)* merupakan gambar Candi Jiwa, tetapi model mengklasifikasikan Candi Blandongan. Dari 20 data citra Candi Jiwa dan Candi Blandongan, model berhasil memprediksi sebanyak 11 dari 13 kelas positif dan 6 dari 7 kelas negatif. Dari hasil prediksi dilakukan perhitungan akurasi untuk menghitung performa model sebagai berikut.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

$$Accuracy = \frac{11 + 6}{11 + 6 + 2 + 1} \times 100\%$$

$$Accuracy = \frac{17}{20} \times 100\% = 85\%$$

Dari hasil perhitungan akurasi menggunakan *confusion matrix* untuk klasifikasi citra Candi Jiwa dan Candi Blandongan mendapatkan nilai sebesar 85%. Hal ini dapat dikatakan bahwa model akurasi *CNN* dan *SVM* memiliki tingkat akurasi yang baik.

#### IV. KESIMPULAN

Berdasarkan penelitian yang sudah dilakukan untuk klasifikasi Candi Jiwa dan Candi Blandongan dapat disimpulkan sebagai berikut.

1. Berdasarkan penelitian yang telah dilakukan, implementasi algoritma *Convolutional Neural Network (CNN)* dilakukan dengan menggunakan lima konvolusi *layer* dan lima *pooling layer* serta pada *feature learning* menggunakan aktivasi *sigmoid* dengan jumlah parameter model sebesar 2.621.618 parameter, kemudian implementasi algoritma *Support Vector Machine (SVM)* dilakukan dengan menggunakan hasil ekstraksi *feature CNN* untuk dijadikan sebagai *inputan SVM*, untuk mengklasifikasikan gambar Candi Jiwa dan Candi Blandongan.

2. Pada pengujian model *CNN* dan *SVM* berhasil dibuat serta mendapatkan nilai akurasi *training* sebesar 83%, akurasi *testing* 85%, dan akurasi validasi sebesar 75%. Kemudian dilakukan evaluasi terhadap model yang telah dibuat menggunakan *confusion matrix*, dengan menggunakan data gambar Candi Jiwa dan Candi Blandongan, dari data citra candi yang tidak digunakan didalam proses *training* dan *testing* model sebelumnya mendapatkan akurasi sebesar 85%. Sehingga dapat disimpulkan bahwa model yang dihasilkan oleh *CNN* dan *SVM* cukup baik.

#### PENGAKUAN

Naskah Ilmiah ini adalah sebagian dari penelitian Tugas Akhir milik Hus'am Badruzzaman, dengan judul "Deteksi Candi Jiwa dan Candi Blandongan Menggunakan Metode *Convolutional Neural Network* dan *Support Vector Machine*" yang dibimbing oleh Tohirin Al Mudzakir dan Rahmat.

#### DAFTAR PUSTAKA

- [1] A. P. Septiawan, L. Agung, dan S. Djono, "Internalisasi Nilai-Nilai Relief Candi Penataran Dalam Pembelajaran Sejarah Untuk Menumbuhkan Kesadaran Budaya," *J. Teor. dan Praksis Pembelajaran IPS*, vol. 3, no. 2, hal. 103–108, 2018, doi: 10.17977/um022v3i22018p103.
- [2] R. Tiarawanti, Yatmin, dan S. Widiatmoko, "Upaya Melestarikan Candi Tegowangi Sebagai Tempat Peninggalan Bersejarah di Kediri," hal. 716–721, 2022.
- [3] BPCB Banten, "Candi Blandongan, salah satu yang unik dari Situs Batujaya," *BPCB Banten*. <https://kebudayaan.kemdikbud.go.id/bpcbbanten/candi-blandongan-salah-satu-yang-unik-dari-situs-batujaya-2/> (diakses 10 Agustus 2023).
- [4] BPCB Banten, "Candi Jiwa," *BPCB Banten*, 2014. <https://kebudayaan.kemdikbud.go.id/bpcbbanten/candi-jiwa/> (diakses 10 Agustus 2023).
- [5] K. A. Baihaqi, Z. Candra, dan B. Nugraha, "Pengenalan Jenis Candi Berdasarkan Bentuk dan Modelnya Menggunakan Metode Convolutional Neural Network (CNN) Pada YOLLO v3," *Syntax J. Inform.*, vol. 10, no. 02, hal. 13–23, 2021.
- [6] Y. Rizki, R. Hayami, dan E. Rahmadani, "Identifikasi Objek Cagar Budaya Candi Mahligai Berbasis Citra Menggunakan Mask R-CNN," *J. Comput. Sci. Inf. Technol.*, vol. 3, no. 3, hal. 309–314, 2022.
- [7] W. Wasis, M. Fachrie, S. Edy, dan Y. Wachid, "Sistem Klasifikasi Relief Candi Borobudur Menggunakan Local Binary Pattern Dan Algoritma Nearest Neighbors," *Vocat. Educ. Natl. Semin. Pap.*, vol. 01, no. 01, hal. 142–147, 2022.
- [8] A. F. P. Suseta dan E. Ade, "Visualisasi Candi Tawang Alun Berbasis Augmented Reality Sebagai Representasi Peninggalan Purba Di Sedati Sidoarjo," *Procedia Eng. Life Sci.*, vol. 2, no. 1, 2021.
- [9] M. A. Henla, "Kajian perbandingan model 3 dimensi menggunakan teknologi terrestrial laser scanner (tls) leica rtc360 dan leica blk2go," 2023.
- [10] W. D. Sulistyono *et al.*, "Bakti Dan Pengenalan Situs Candi Kalicilik Pada Siswa SD 1 Candirejo Ponggok Blitar," *J. Praksis dan Dedik. Sos.*, vol. 2, no. 2, hal. 54–61, 2019, doi: 10.17977/um032v2i2p54-61.
- [11] Y. N. Nabusa, "Pengolahan Citra Digital Perbandingan Metode Histogram Equalization Dan Spesification Pada Citra Abu-Abu," *J-Icon*, vol. 7, no. 1, hal. 87–95, 2019.
- [12] A. Peryanto, A. Yudhana, dan R. Umar, "Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation," *J. Appl. Informatics Comput.*, vol. 4, no. 1, hal. 45–51, 2020, doi: 10.30871/jaic.v4i1.2017.
- [13] F. M. Hana, "Klasifikasi Penderita Penyakit Diabetes Menggunakan Algoritma Decision Tree C4.5," *J. Sist. Komput. dan Kecerdasan Buatan*, vol. 4, no. 1, hal. 32–39, 2020.
- [14] M. A. Pangestu dan H. Bunyamin, "Analisis Performa dan Pengembangan Sistem Deteksi Ras Anjing pada Gambar dengan Menggunakan Pre-Trained CNN Model," *J. Tek. Inform. dan Sist. Inf.*, vol. 4, hal. 337–344, 2018, doi: <http://dx.doi.org/10.28932/jutisi.v4i2.828>.
- [15] R. D. Nurfiti, "Implementasi Deep Learning Berbasis Tensorflow Untuk Pengenalan Sidik Jari," *J. Emit.*, vol. 18, no. 01, hal. 22–27, 2018.
- [16] Hermanto, A. Mustopa, dan A. Y. Kuntoro, "Algoritma Klasifikasi Naive Bayes Dan Support Vector Machine Dalam Layanan Komplain Mahasiswa," *JITK (Jurnal Ilmu Pengetah. dan Teknol. Komputer)*, vol. 5, no. 2, hal. 211–220, 2020, doi: 10.33480/jitk.v5i2.1181.
- [17] R. Rosdiana, M. Ula, dan H. A. K. Aidilof, "Implementasi Pemodelan Citra Model Svm (Support Vector Machine) Dalam Penentuan Pengklasifikasian Jenis Suara Kontes Burung," *J. Inform. Kaputama*, vol. 5, no. 2, 2021, [Daring]. Tersedia pada: <http://jurnal.kaputama.ac.id/index.php/JIK/article/view/566>
- [18] S. Ahlawat dan A. Choudhary, "Hybrid CNN-SVM Classifier for Handwritten Digit Recognition," *Procedia Comput. Sci.*, vol. 167, no. 2019, hal. 2554–2560, 2020, doi: 10.1016/j.procs.2020.03.309.
- [19] D. Normawati dan S. A. Prayogi, "Implementasi Naive Bayes Classifier Dan Confusion Matrix Pada Analisis Sentimen Berbasis Teks Pada Twitter," *J. Sains Komput. Inform. (J-SAKTI)*, vol. 5, no. 2, hal. 697–711, 2021.
- [20] Yohannes, D. Udjulawa, dan Febbiola, "Klasifikasi Lukisan Karya Van Gogh Menggunakan Convolutional Neural Network-Support Vector Machine," *J. Tek. Inform. dan Sist. Inf.*, vol. 7, no. 1, hal. 192–205, 2021, doi: 10.28932/jutisi.v7i1.3399.