LAPORAN

KLASIFIKASI KUCING DAN ANJING MENGGUNAKAN SUPPORT VECTOR MACHINE (SVM)

Disusun Oleh Kelompok 7:

Wiwik Ainun Janah	230411100017
Siti Khoirul Muzaro'ah	230411100068
Fatimah Azzahra	230411100015
Yasmin Azzahra	230411100186

Mata Kuliah :

Pengolahan Citra

Dosen Pengampu :

Prof. Dr. Rima Tri Wahyuningrum, ST., MT.

PRODI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS TRUNOJOYO MADURA

2025

DAFTAR ISI

BAB I
PENDAHULUAN

1.1 Latar Belakang

Perkembangan teknologi pengolahan citra (image processing) memungkinkan komputer melakukan proses identifikasi objek secara otomatis. Salah satu permasalahan umum yang banyak dikembangkan adalah klasifikasi gambar anjing dan kucing, karena keduanya memiliki karakteristik visual yang mirip dalam berbagai kondisi pencahayaan, sudut pandang, maupun pose. Klasifikasi otomatis ini dapat digunakan dalam berbagai aplikasi, seperti sistem pencarian hewan hilang, identifikasi hewan peliharaan, manajemen database shelter, hingga sistem pengawasan berbasis kamera.

Untuk menyelesaikan permasalahan tersebut, dibutuhkan algoritma machine learning yang mampu bekerja baik pada data citra berukuran besar dan memiliki variasi tekstur serta bentuk yang beragam. Metode Support Vector Machine (SVM) merupakan salah satu algoritma klasifikasi yang dikenal memiliki performa tinggi dalam memisahkan data antar kelas dengan hyperplane terbaik. SVM sangat efektif digunakan pada data hasil ekstraksi fitur, khususnya fitur tekstur seperti Histogram of Oriented Gradient (HOG).

Dalam penelitian ini diterapkan metode SVM untuk melakukan klasifikasi citra anjing dan kucing. Proses dimulai dengan preprocessing berupa resize gambar, konversi ke grayscale, dan ekstraksi fitur menggunakan metode HOG serta flatten pada kanal warna. Setelah itu dilakukan normalisasi menggunakan StandardScaler dan pemodelan menggunakan SVM kernel linear. Penelitian ini diharapkan mampu menghasilkan model klasifikasi yang akurat dan efisien.

1.2 Rumusan Masalah

Rumusan masalah penelitian ini adalah:

1. Bagaimana proses preprocessing dan ekstraksi fitur pada citra anjing dan kucing menggunakan metode HOG?
2. Bagaimana penerapan algoritma Support Vector Machine (SVM) dalam klasifikasi citra anjing dan kucing?
3. Bagaimana performa model SVM kernel linear terhadap dataset gambar anjing dan kucing?

1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu:

1. Melakukan preprocessing citra berupa resize, grayscale, dan ekstraksi fitur menggunakan HOG serta flatten kanal warna.
2. Menerapkan algoritma Support Vector Machine (SVM) untuk klasifikasi gambar anjing dan kucing.

3. Mengevaluasi performa model menggunakan metrik evaluasi seperti accuracy, precision, recall, dan confusion matrix.

BAB II
TINJAUAN PUSTAKA

2.1 Pengolahan Citra Digital

Pengolahan Citra Digital (PCD) merupakan cabang ilmu yang mempelajari teknik-teknik pemrosesan, analisis, dan transformasi citra menggunakan perangkat komputasi. Tujuan utamanya adalah meningkatkan kualitas visual citra, mengekstraksi informasi yang relevan, serta mempersiapkan citra untuk tahap analisis lebih lanjut, seperti klasifikasi atau pengenalan objek.

a. Citra Digital dan Representasi

   Citra digital direpresentasikan sebagai matriks dua dimensi yang tersusun dari elemen-elemen diskrit bernama piksel. Setiap piksel memiliki nilai intensitas atau warna tertentu yang menggambarkan informasi visual pada titik tersebut. Pada citra berwarna (RGB), setiap piksel terdiri dari tiga kanal warna—merah, hijau, dan biru—yang masing-masing memiliki nilai intensitas antara 0 hingga 255. Sementara itu, citra grayscale hanya memiliki satu kanal dengan rentang nilai intensitas yang sama, sehingga lebih sederhana untuk dianalisis secara komputasi.

b. Jenis Citra Berdasarkan Kedalaman Warna

Jenis Citra	Deskripsi	Implementasi dalam Code
Grayscale	Satu kanal warna (0–255)	Citra RGB diubah ke grayscale sebelum HOG
Warna (RGB)	3 channel: Red, Green, Blue	Dataset awal berupa citra RGB

2.2 Ekstraksi Fitur dalam Pengolahan Citra

Ekstraksi fitur merupakan tahap penting dalam analisis citra, karena bertujuan mengubah informasi visual dalam bentuk piksel menjadi representasi numerik yang lebih bermakna. Representasi ini dikenal sebagai vektor fitur, yang berisi karakteristik tertentu seperti bentuk, tekstur, atau pola tepi. Salah satu metode yang banyak digunakan dalam ekstraksi fitur adalah Histogram of Oriented Gradients (HOG).

a. Histogram of Oriented Gradients (HOG)

   HOG merupakan teknik yang menekankan pada struktur dan bentuk objek melalui analisis pola gradien. Metode ini bekerja dengan menghitung arah dan besarnya perubahan intensitas antar-piksel, kemudian menyusunnya ke dalam histogram orientasi gradien pada area-area kecil citra. Pendekatan ini efektif untuk mendeskripsikan kontur objek dan tidak sensitif terhadap variasi pencahayaan, sehingga banyak digunakan dalam deteksi dan klasifikasi objek.

2.3 Support Vector Machine (SVM)

Support Vector Machine (SVM) adalah algoritma pembelajaran terawasi yang digunakan secara luas dalam permasalahan klasifikasi maupun regresi. Pada permasalahan klasifikasi, SVM bertujuan menemukan sebuah bidang pemisah (hyperplane) yang mampu membedakan data dari berbagai kelas secara optimal.

a. Hyperplane dan Margin Optimal
   SVM menentukan sebuah hyperplane yang memisahkan data ke dalam dua kelas atau lebih. Pemilihan hyperplane optimal dilakukan dengan memaksimalkan margin, yaitu jarak antara hyperplane dan titik data terdekat dari masing-masing kelas. Titik-titik data yang berada pada area margin disebut sebagai support vectors, dan titik-titik inilah yang paling berpengaruh dalam menentukan posisi hyperplane. Konsep margin yang besar berkontribusi pada kemampuan generalisasi model yang lebih baik.
b. Kernel Trick dalam SVM
   SVM dapat menangani data non-linear melalui penggunaan teknik kernel (kernel trick). Kernel melakukan pemetaan data ke ruang fitur berdimensi lebih tinggi sehingga data yang semula sulit dipisahkan secara linear menjadi dapat dipisahkan pada ruang tersebut. Beberapa kernel yang umum digunakan antara lain kernel linear, polynomial, dan radial basis function (RBF). Kernel RBF sering dipilih karena kemampuannya dalam memodelkan pola data yang kompleks dan non-linear.
c. Kelebihan dan Kekurangan Support Vector Machine (SVM)

Aspek	Kelebihan	Kekurangan
Kemampuan Generalisasi	Memiliki margin yang besar sehingga mampu melakukan generalisasi dengan baik pada data baru.	Rentan terhadap penurunan performa jika parameter tidak dipilih secara optimal.
Kinerja pada Dimensi Tinggi	Sangat efektif digunakan pada ruang fitur berdimensi tinggi, seperti pada kasus pengolahan citra.	Membutuhkan waktu komputasi tinggi ketika jumlah data sangat besar.
Ketergantungan pada Data	Hanya bergantung pada *support vectors*, sehingga model lebih efisien dibandingkan metode yang menggunakan seluruh data.	Sensitif terhadap data outlier yang dapat memengaruhi posisi hyperplane secara signifikan.
Kemampuan Menangani Non-Linearitas	Kernel trick memungkinkan pemisahan data	Pemilihan kernel dan parameter seperti C dan gamma cukup

	non-linear ke dalam ruang dimensi lebih tinggi.	kompleks dan membutuhkan proses optimasi.
Risiko Overfitting	Memiliki risiko overfitting yang rendah pada banyak kasus, terutama ketika parameter regulasi ditetapkan dengan tepat.	Kurang efektif jika data antar kelas sangat overlapping sehingga hyperplane sulit ditentukan.
Output Prediksi	Memberikan keputusan klasifikasi yang tegas berdasarkan batas pemisah optimal.	Tidak menghasilkan probabilitas secara langsung; memerlukan metode tambahan untuk konversi probabilitas.

2.4 Tahapan Umum Klasifikasi Citra

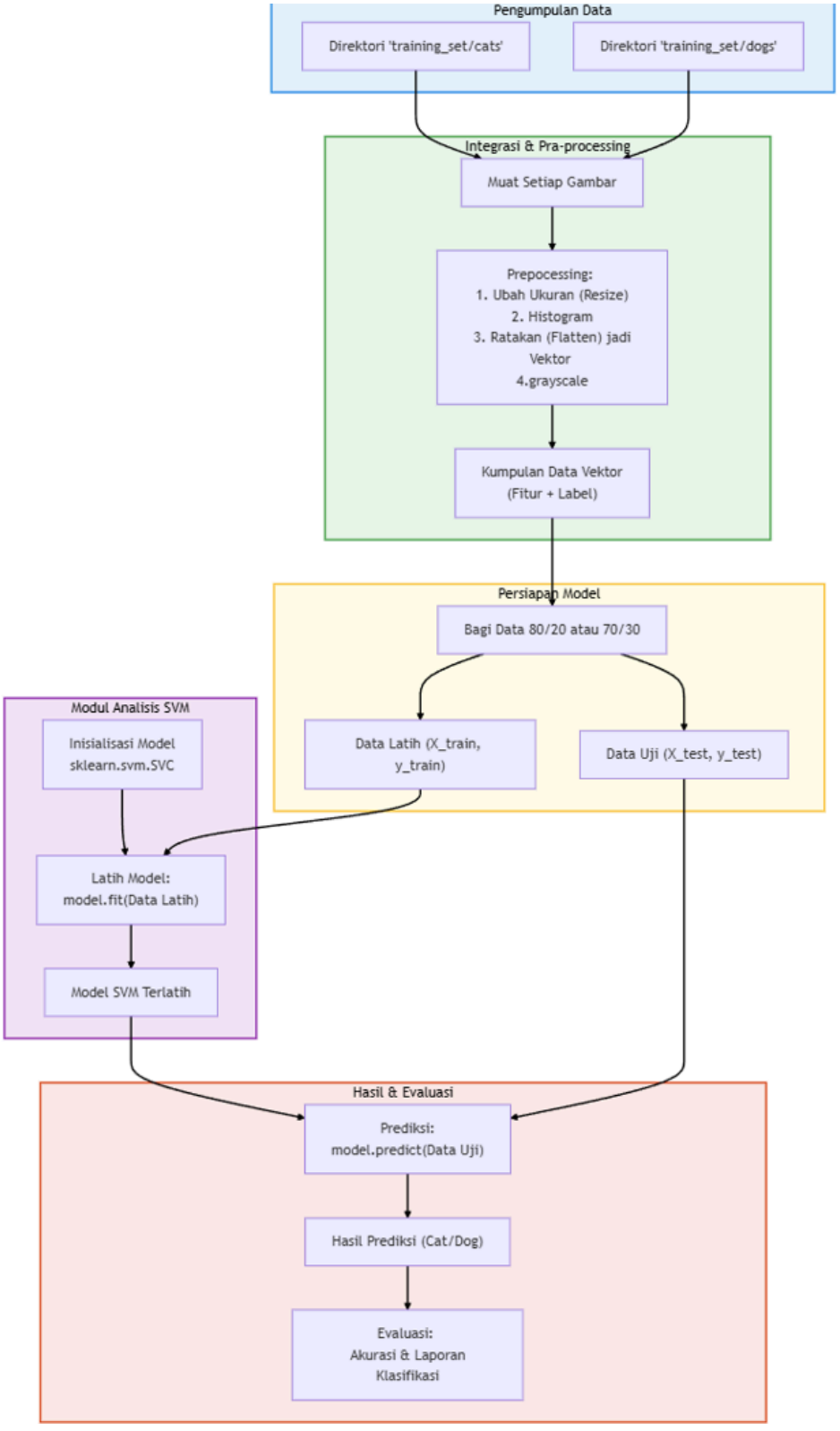
Klasifikasi citra merupakan proses yang bertujuan mengelompokkan citra ke dalam kategori tertentu berdasarkan karakteristik yang dimilikinya. Secara umum, proses klasifikasi citra melibatkan empat tahapan utama sebagai berikut:

a. Dataset dan Pra-Pengolahan Citra
   Tahap ini mencakup proses pengumpulan citra dari sumber tertentu serta persiapan citra sebelum dianalisis lebih lanjut. Beberapa langkah yang termasuk di dalamnya antara lain penyeragaman ukuran (resizing), peningkatan kualitas visual (enhancement), konversi ruang warna (misalnya menjadi grayscale), dan segmentasi objek apabila diperlukan. Pra-pengolahan bertujuan menghasilkan citra yang bersih dan konsisten sehingga mempermudah proses ekstraksi fitur.
b. Ekstraksi Fitur
   Pada tahap ini, citra diubah menjadi vektor fitur yang merepresentasikan karakteristik tertentu dari citra tersebut. Metode ekstraksi fitur dapat berfokus pada bentuk, tekstur, pola tepi, maupun karakteristik warna. Representasi fitur yang baik sangat penting karena menjadi dasar bagi model klasifikasi untuk mengenali pola antar kelas.
c. Pelatihan Model Klasifikasi
   Vektor fitur yang dihasilkan dari tahap sebelumnya digunakan sebagai input untuk melatih model klasifikasi. Proses ini melibatkan algoritma pembelajaran terawasi seperti SVM yang mempelajari pola hubungan antara fitur dan label kelas. Pembagian data biasanya dilakukan menjadi data latih dan data uji untuk memastikan bahwa model mampu melakukan generalisasi terhadap data baru.
d. Pengujian dan Evaluasi

Tahap ini bertujuan menilai performa model klasifikasi dengan menggunakan data uji yang tidak digunakan pada saat pelatihan. Evaluasi dapat dilakukan dengan berbagai metrik seperti akurasi, presisi, recall, F1-score, serta analisis confusion matrix. Metrik-metrik tersebut memberikan gambaran mengenai kemampuan model dalam mengklasifikasikan citra secara benar.

BAB III
METODOLOGI

3.1 Arsitektur Sistem



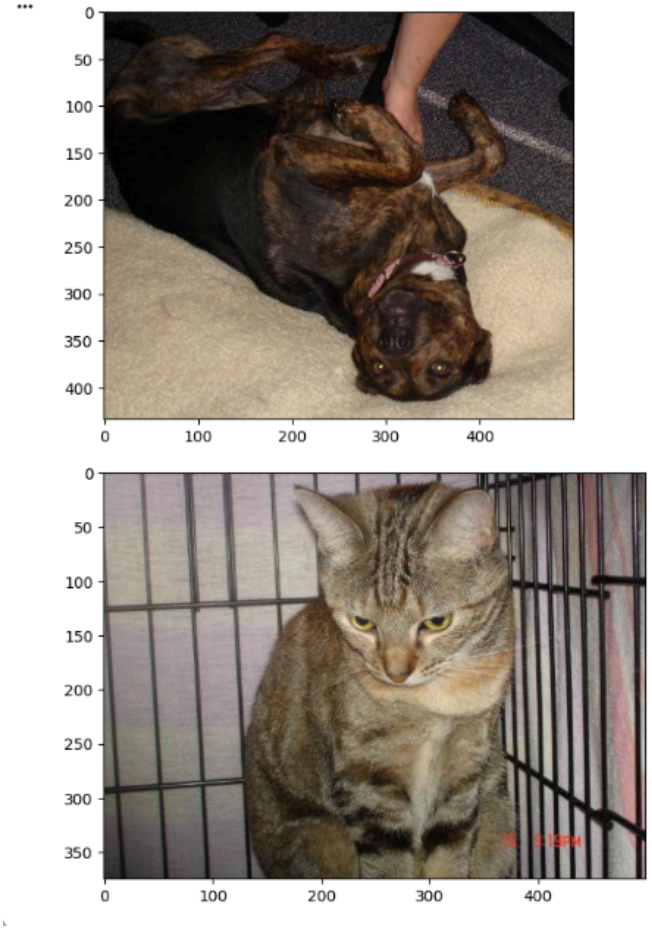
a. Pengumpulan Data

Langkah pertama adalah mengumpulkan data gambar yang akan digunakan untuk melatih model. Pada diagram, data berasal dari dua direktori:

- training_set/cats untuk menyimpan gambar kucing.
- training_set/dogs untuk menyimpan gambar anjing.

Tujuannya agar data terorganisir sesuai dengan labelnya. Dengan struktur folder ini, program dapat dengan mudah membaca dan memberi label otomatis pada gambar—misalnya, semua gambar di folder *cats* akan diberi label "0" (kucing), dan semua gambar di folder *dogs* diberi label "1" (anjing).
sumber data:
https://www.kaggle.com/datasets/tongpython/cat-and-dog

```
def get_image(path):
    img = Image.open(path)
    return np.array(img)

# showing a dog image
dog_row = labels_df[labels_df.label == 1].reset_index().image[23]
plt.imshow(get_image(dog_row))
plt.show()

# showing a cat image
cat_row = labels_df[labels_df.label == 0].reset_index().image[79]
plt.imshow(get_image(cat_row))
plt.show()
```



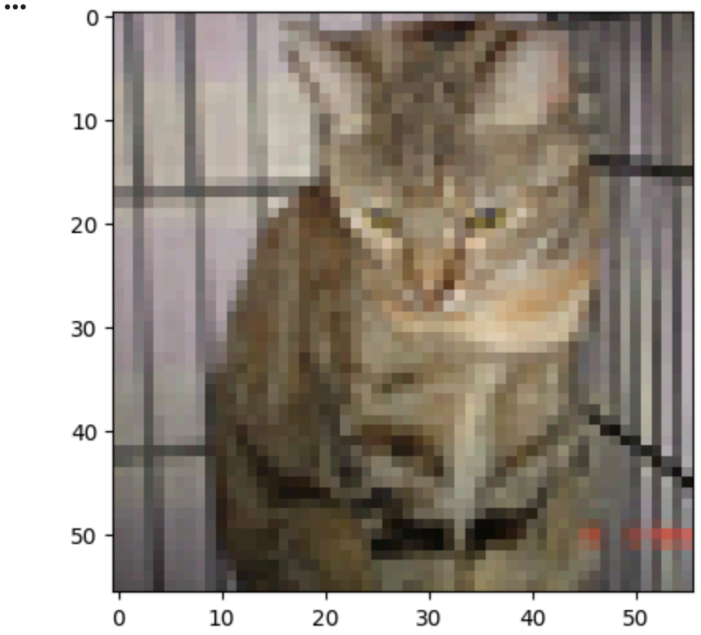
b. Integrasi & Pra-Processing

Setelah data terkumpul, setiap gambar dimuat dan diproses agar memiliki bentuk dan ukuran yang seragam sebelum masuk ke tahap pelatihan. Tahapan pra-proses yang dilakukan yaitu:

1. Ubah ukuran (Resize)

   Semua gambar disesuaikan ke ukuran yang sama (misalnya 128x128 piksel) agar model tidak bingung oleh perbedaan resolusi.

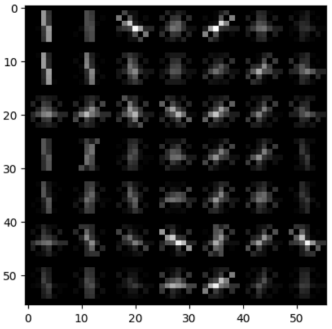
```
# image manupulation
img = Image.open(cat_row)
img = resizeimage.resize_cover(img, [56, 56])
plt.imshow(np.array(img), cmap='gray')
plt.show()
```



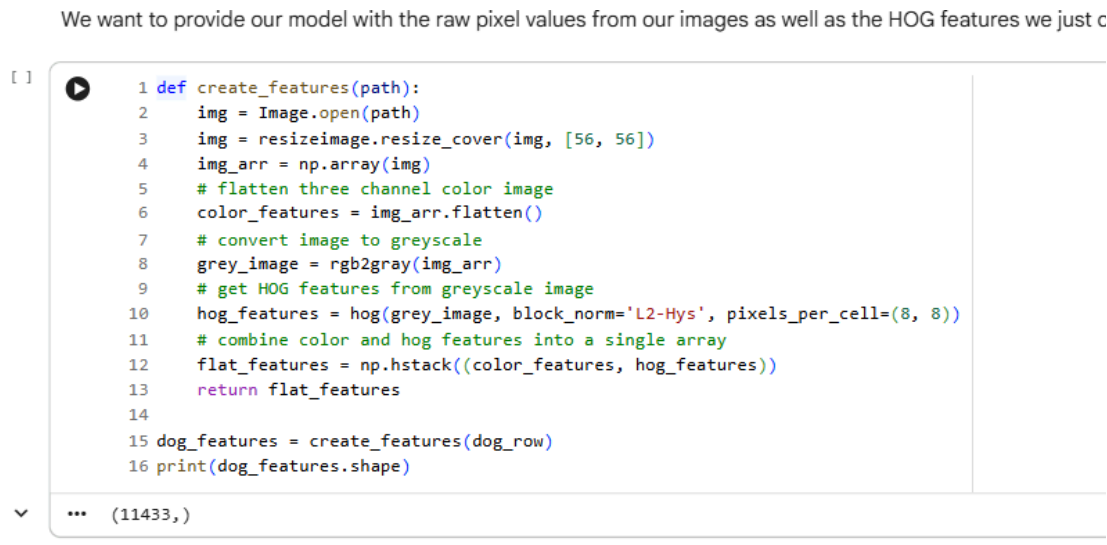
2. Histogram of oriented gradient

   Dapat digunakan untuk mengekstraksi distribusi intensitas warna atau tingkat kecerahan gambar, yang bisa menjadi fitur pembeda antara gambar kucing dan anjing.

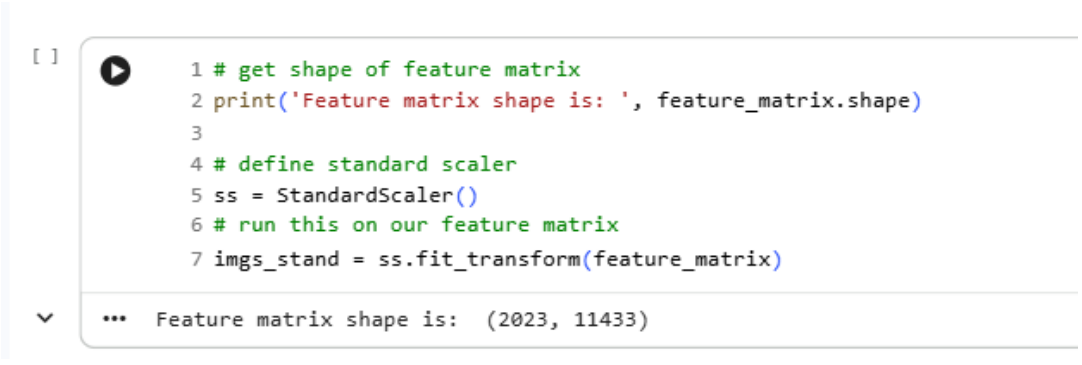
```
grey_img = rgb2gray(img)
hog_features, hog_img = hog(grey_img, visualize=True, block_norm='L2-Hys', pixels_per_cell=(8,8))
plt.imshow(np.array(hog_img), cmap='gray')
plt.show()
```



3. Ratakan (Flatten)
   Setelah diubah ukurannya, gambar biasanya berbentuk matriks 2D atau 3D (misalnya 128x128x3). Matriks ini diubah menjadi vektor satu dimensi agar bisa diproses oleh algoritma SVM.
4. Grayscale
   Jika warna tidak diperlukan, gambar diubah menjadi hitam-putih (grayscale) untuk menyederhanakan data dan mengurangi kompleksitas perhitungan.



5. standarisasi



Setelah semua proses di atas, dihasilkan data vektor fitur dan labelnya. Data inilah yang akan digunakan untuk pelatihan dan pengujian model.

c. Persiapan Model
   Tahap ini membagi data hasil pra-proses menjadi dua bagian:
   - Data Latih: digunakan untuk melatih model agar mengenali pola.
   - Data Uji: digunakan untuk mengukur seberapa baik model memprediksi data baru yang belum pernah dilihat.

   Umumnya digunakan pembagian seperti 80/20 (80% data latih, 20% data uji) atau 70/30 tergantung banyaknya data yang tersedia.

Langkah ini penting agar hasil evaluasi nanti menggambarkan kemampuan model secara objektif, bukan hanya kemampuan mengingat data latih.

```
X_train, X_test, y_train, y_test = train_test_split(imgs_stand,
                                                    labels_df.label.values,
                                                    test_size=0.3,
                                                    random_state=1234123)

# look at the distrubution of labels in the train set
pd.Series(y_train).value_counts()
```

	count
0	714
1	702

dtype: int64

d. Modul Analisis SVM (Support Vector Machine)

Setelah data dibagi, tahap ini mempersiapkan model SVM yang akan digunakan untuk klasifikasi gambar.

Langkah-langkahnya:

1. Inisialisasi Model
   Membuat model SVM dengan parameter tertentu (misalnya jenis kernel linear, polynomial, atau RBF).
2. Latih Model
   Model dilatih dengan data latih.
3. Pada proses ini, SVM mencari *hyperplane terbaik* yang memisahkan dua kelas (kucing dan anjing) seakurat mungkin.
4. Model Terlatih
   Setelah pelatihan selesai, model SVM yang siap digunakan untuk melakukan prediksi terhadap data baru.

Train model

Now let's finally build our model! We'll do this using an SVM classifier with a linear kernel.

```
# define support vector classifier
svm = SVC(kernel='linear',probability=True, random_state=42)

# fit model
svm.fit(X_train, y_train)
```

```
SVC(kernel='linear', probability=True, random_state=42)
```

e. Hasil & Evaluasi

Tahap terakhir adalah menggunakan model yang telah dilatih untuk membuat prediksi dan mengevaluasi kinerjanya.

- Prediksi

  Model digunakan untuk memprediksi kelas data uji:

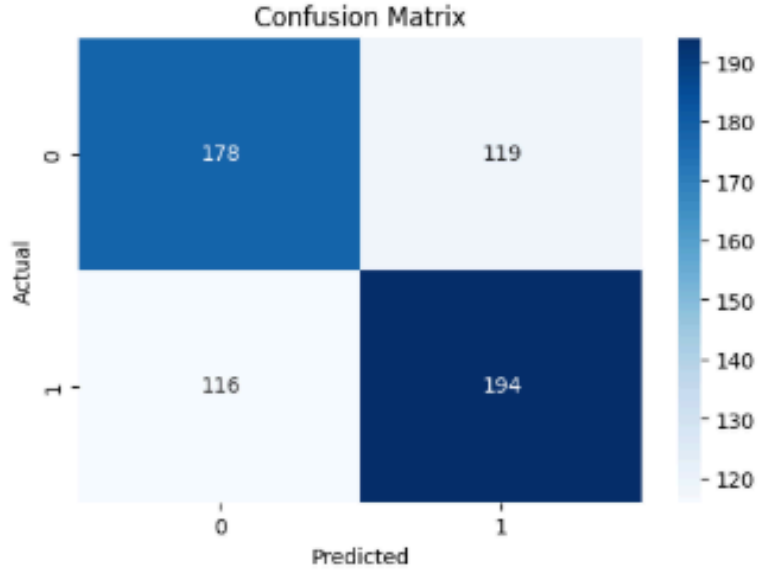
  1. Output-nya adalah hasil klasifikasi (cat/dog) untuk setiap gambar uji.
  2. Evaluasi

     Hasil prediksi dibandingkan dengan label asli untuk menghitung:

     - Akurasi → seberapa banyak prediksi yang benar.
     - Precision, Recall, F1-score → metrik evaluasi yang lebih rinci.
     - Laporan klasifikasi atau confusion matrix → untuk melihat jenis kesalahan yang sering terjadi.

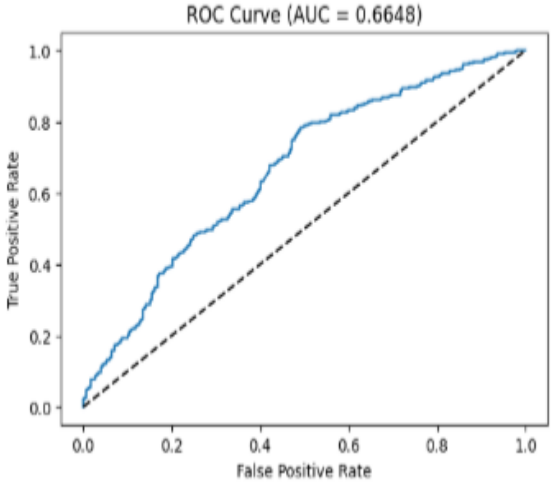
```python
from sklearn.metrics import (
    accuracy_score, classification_report, confusion_matrix,
    roc_auc_score, roc_curve, precision_score, recall_score, f1_score
)
import matplotlib.pyplot as plt
import seaborn as sns

# ============================
# 1. Generate Predictions
# ============================
y_pred = svm.predict(X_test)

# Jika model support probability
try:
    y_prob = svm.predict_proba(X_test)[:, 1]
except:
    # Untuk SVM kernel non-linear tanpa probability -> gunakan decision function
    y_prob = svm.decision_function(X_test)

# ============================
# 2. Akurasi
# ============================
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)

# ============================
# 3. Precision, Recall, F1-score
# ============================
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

TN, FP, FN, TP = cm.ravel()

specificity = TN / (TN + FP)

print("\nConfusion Matrix:\n", cm)
print("Specificity:", specificity)

# Plot Confusion Matrix
plt.figure(figsize=(6, 4))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues")
plt.title("Confusion Matrix")
plt.xlabel("Predicted")
plt.ylabel("Actual")
plt.show()

# ============================
# 6. ROC Curve + AUC
# ============================
auc = roc_auc_score(y_test, y_prob)
fpr, tpr, thresholds = roc_curve(y_test, y_prob)

print("AUC Score:", auc)

plt.figure(figsize=(6, 4))
plt.plot(fpr, tpr)
plt.plot([0, 1], [0, 1], 'k--')
plt.xlabel("False Positive Rate")
plt.ylabel("True Positive Rate")
plt.title("ROC Curve (AUC = {:.4f})".format(auc))
plt.show()
```

```
Accuracy: 0.6128500823723229

Precision: 0.6198083067092651
Recall: 0.6258064516129033
F1-score: 0.622792937399679

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

           0       0.61      0.60      0.60       297
           1       0.62      0.63      0.62       310

    accuracy                           0.61       607
   macro avg       0.61      0.61      0.61       607
weighted avg       0.61      0.61      0.61       607


Confusion Matrix:
 [[178 119]
 [116 194]]
Specificity: 0.5993265993265994
```




```
AUC Score: 0.6647768002606712
```

AUC Score: 0.6647768002606712



Visualizing Predictions

Let's take random images from the test set and predict and display them individually. Here's we'll realize how bad this model is TwT

```
from random import randint

test = ImageFolder("/content/drive/MyDrive/cats and dogs/")
imgs, labels = zip(*test.imgs)
imgs = list(imgs)
labels = list(labels)
```

```
random_ix = randint(0, len(imgs))
label = {0: 'Cat', 1:'Dog'}
rand_img = imgs[random_ix]
# create features of the image
test_features = create_features(rand_img)
# predict
prediction = svm.predict([test_features])
print("Prediction: " + label[prediction[0]])
print("Actual: " + label[labels[random_ix]])
# display image
display(Image.open(rand_img))
```

```
Prediction: Cat
Actual: Cat
```

BAB IV
KESIMPULAN

4.1 Kesimpulan

Berdasarkan penelitian mengenai klasifikasi citra kucing dan anjing menggunakan metode Support Vector Machine (SVM) dengan ekstraksi fitur Histogram of Oriented Gradient (HOG), dapat disimpulkan bahwa:

1. Tahap preprocessing berupa resize citra, konversi grayscale, ekstraksi fitur HOG, dan flatten vektor fitur berhasil mengubah citra menjadi data numerik yang siap digunakan dalam proses klasifikasi.
2. Metode HOG mampu menangkap pola bentuk dan tepi objek pada citra, sehingga fitur yang dihasilkan cukup efektif untuk membedakan karakteristik visual antara gambar kucing dan anjing.
3. Algoritma Support Vector Machine kernel linear mampu melakukan proses klasifikasi dengan performa yang cukup baik terhadap dataset, meskipun hasil akurasinya belum sempurna karena kompleksitas variasi citra seperti pencahayaan, posisi objek, dan latar belakang gambar.
4. Evaluasi menggunakan metrik seperti accuracy, precision, recall, dan confusion matrix menunjukkan bahwa model dapat mengenali sebagian besar gambar dengan benar, sehingga metode ini layak digunakan pada tugas klasifikasi citra berskala kecil hingga menengah.
5. Secara umum, kombinasi HOG + SVM dapat menjadi alternatif solusi klasifikasi citra berbasis bentuk (shape-based classification) khususnya ketika sumber daya komputasi terbatas dan penggunaan model Deep Learning belum memungkinkan.

DOKUMENTASI