

Pengenalan Pola (KCV)
Program Pendeteksian Objek Stik dengan OpenCV



DOSEN PENGAMPU :
Dr. Feri Chandra, S.T., M.T.

DISUSUN OLEH :
Reno Widi Respati (2007110668)
Levita Wahyuni (2007125741)

PRODI TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS RIAU
2022

KATA PENGANTAR

Puji dan syukur kami panjatkan kehadiran Allah SWT karena berkat karunia, rahmat, dan nikmat-Nya kami dapat menyelesaikan Tugas proyek UAS ini. Makalah ini kami buat untuk memenuhi salah satu tugas dari mata kuliah Pengenalan Pola (KCV), dengan judul proyek Program Pendeteksian Objek Stik Dengan Opencv. Semoga tugas makalah yang telah kami buat ini dapat diterima dan bermanfaat bagi para pembacanya. Terlepas dari semua itu, kami menyadari sepenuhnya bahwa masih ada kekurangan baik dari segi susunan kalimat maupun tata bahasanya.

Oleh karena itu dengan tangan terbuka kami menerima segala saran dan kritik dari pembaca agar kami dapat memperbaiki makalah ini. Akhir kata kami ucapkan terima kasih kepada semua yang telah membantu dan berharap semoga makalah ini dapat bermanfaat dan dapat memberikan inspirasi terhadap pembaca.

Pekanbaru, 18 Desember 2022

Penyusun

DAFTAR ISI

| | |
|-----------------------------------|-----|
| COVER..... | i |
| KATA PENGANTAR..... | ii |
| DAFTAR ISI | iii |
| BAB I PENDAHULUAN | 1 |
| 1.1. Latar Belakang..... | 1 |
| 1.2. Rumusan Masalah..... | 1 |
| 1.3. Tujuan..... | 2 |
| BAB II LANDASAN TEORI..... | 3 |
| 2.1. Python..... | 3 |
| 2.2. OpenCV | 3 |
| 2.3. Visual Studio Code..... | 3 |
| 2.4. Cascade Trainer GUI..... | 4 |
| 2.5. Command Prompt..... | 4 |
| BAB III CARA KERJA APLIKASI | 5 |
| 3.1. Algoritma Classier | 5 |
| 3.2. Diagram Blok..... | 6 |
| BAB IV HASIL DAN PEMBAHASAN..... | 7 |
| 4.1. Hasil..... | 7 |
| 4.2. Pembahasan | 7 |
| BAB V PENUTUP | 9 |
| 5.1. Kesimpulan | 9 |
| 5.2. Saran | 9 |
| DAFTAR PUSTAKA..... | 10 |
| LAMPIRAN | 11 |

BAB I

PENDAHULUAN

1.1. Latar Belakang

Seperti yang kita ketahui, Perkembangan teknologi yang sangat pesat di era globalisasi saat ini telah memberikan banyak manfaat dalam kemajuan diberbagai aspek sosial. Penggunaan teknologi oleh manusia dalam membantu menyelesaikan pekerjaan merupakan hal yang menjadi keharusan dalam kehidupan.

Maraknya penggunaan komputer di berbagai bidang kehidupan manusia, menuntut setiap orang untuk mengetahui dan mempelajari berbagai macam software pendukung yang dapat beroperasi pada komputer tersebut. Kemampuan komputer dalam melakukan penghitungan yang sangat cepat, dapat mempermudah manusia atau user dalam mengoperasikannya. Berbagai macam media pendukung untuk mengoptimalkan kinerja komputer banyak dibuat. Media tersebut berupa software atau program aplikasi yang hubungannya tidak dapat terpisahkan dari komputer.

Python adalah bahasa pemrograman model skrip (*scripting language*) yang berorientasi obyek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya.

Sehubungan dengan hal diatas pada pembuatan tugas akhir ini akan dibuat sebuah program pendeteksian objek stik dengan opencv menggunakan aplikasi Visual Studio Code dan menggunakan *library* OpenCV dan Python.

1.2. Rumusan Masalah

Pendeteksi objek stik dapat dirumuskan persoalannya sebagai berikut :

1. Bagaimana membangun sistem pendeteksian objek stik dengan OpenCV dan Python.
2. Bagaimana karakter objek stik yang dapat dideteksi oleh sistem.

1.3. Tujuan

1. Dapat membuat suatu sistem sederhana yang dapat mendeteksi objek stik game dengan menggunakan pemrograman Python dan diterapkan pada OpenCV
2. Dapat mengetahui objek stik yang dapat terdeteksi oleh sistem.

BAB II

LANDASAN TEORI

2.1. Python

Python merupakan salah satu bahasa pemrograman tingkat tinggi yang bersifat interpreter, interaktif, object-oriented, dan dapat beroperasi di hampir semua platform, seperti keluarga UNIX, Mac, Windows, dan lainnya.[1] Python salah satu bahasa pemrograman yang banyak digunakan oleh perusahaan besar maupun para developer untuk mengembangkan berbagai macam aplikasi berbasis desktop, web dan mobile.[2] Namun demikian python juga memiliki kekurangan yaitu cukup lambat dijalankan, kurang support terhadap android dan IOS, memiliki keterbatasan dalam akses basis data, dan juga tidak cocok untuk melakukan pekerjaan multi-core/multiprocessor.

Python diciptakan oleh Guido van Rossum di Belanda pada tahun 1990 dan namanya diambil dari acara televisi kesukaan Guido Monty Python's Flying Circus. Van Rossum mengembangkan Python sebagai hobi, kemudian Python menjadi bahasa pemrograman yang dipakai secara luas dalam industri dan pendidikan karena sederhana, ringkas, sintak intuitif dan memiliki pustaka yang luas

2.2. OpenCV

OpenCV (Open Source Computer Vision Library) adalah sebuah library yang digunakan untuk mengolah gambar dan video sehingga pengguna dapat meng-ekstrak informasi dari sebuah citra. OpenCV dikembangkan oleh perusahaan Intel yang berfokus untuk menyederhanakan programming yang terkait dengan citra. OpenCV dapat berjalan dengan berbagai bahasa pemrograman, seperti C, C++, Java, Python, dan juga support diberbagai platform seperti Windows, Linux, Mac OS, iOS dan Android. [3]

2.3. Visual Studio Code

Visual studio code merupakan sebuah teks *editor* ringan dan handal yang dibuat oleh *Microsoft* untuk sistem operasi *multiplatform*, artinya tersedia juga untuk versi *Linux*, *Mac*, dan *Windows*. Teks *editor* ini secara langsung

mendukung bahasa pemrograman *JavaScript*, *Typescript*, dan *Node.js*, serta bahasa pemrograman lainnya dengan bantuan plugin yang dapat dipasang via *marketplace Visual studio code* (seperti C++, C#, Python, Go, Java, dst).[4]

Visual Studio code didasarkan pada *Elektron*, kerangka kerja yang digunakan untuk menyebarkan aplikasi *Node.js* untuk *desktop* yang berjalan pada *Blinklayout*. Meskipun menggunakan kerangka *Elektron*, *Visual studio code* tidak menggunakan Atom dan menggunakan komponen *editor* yang sama (diberi kode nama “*Monaco*”) yang digunakan dalam *Visual Studio Team Services* yang sebelumnya disebut *Visual Studio Online*. [5]

2.4. Cascade Trainer GUI

Cascade Trainer GUI adalah sebuah software yang digunakan untuk melatih sebuah model deteksi menggunakan teknik Haar-like Feature. Aplikasi ini dikembangkan oleh OpenCV, yaitu sebuah library open source yang berisi berbagai alat dan fungsi yang berguna untuk pengolahan citra dan video. Software ini menyediakan interface yang mudah digunakan untuk mengatur dataset yang akan digunakan untuk proses training, serta menampilkan hasil prediksi model secara real-time selama proses training berlangsung.

2.5. Command Prompt

Command Prompt adalah sebuah aplikasi yang digunakan untuk menjalankan perintah-perintah dari sistem operasi Windows melalui antarmuka baris perintah (command-line interface). Command Prompt digunakan untuk mengakses fitur dan fungsi yang tidak tersedia di antarmuka grafis sistem operasi Windows, seperti mengelola file dan folder, mengatur konfigurasi sistem, menjalankan program-program tertentu, dan lain-lain.

BAB III

CARA KERJA APLIKASI

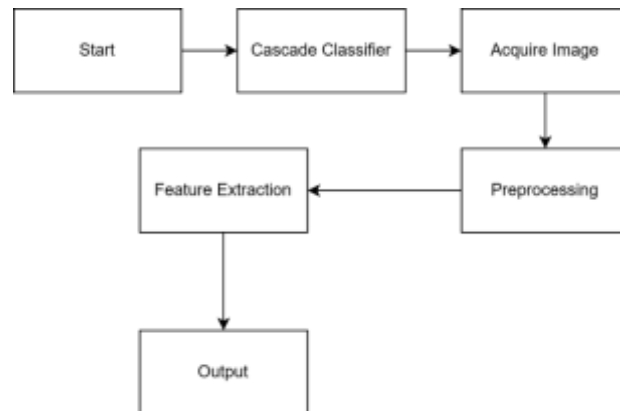
3.1. Algoritma Classier

Program pendeteksi stik game dibuat menggunakan algoritma yang berbasis Cascade Classifier. Program pendeteksi stik game yang menggunakan Cascade Classifier adalah program yang dapat mendeteksi objek secara video lokal ataupun secara realtime. Untuk melakukan pendeteksian objek stik game ini, kita harus mengumpulkan dataset yang terdiri dari gambar positif dan juga gambar negatif. Dataset positif berisi berbagai gambar objek yang ingin dideteksi bentuknya, sedangkan dataset negatif berisi gambar acak yang tidak terdapat gambar objek yang ingin dideteksi.

Setelah mempunyai dataset positif dan negatif, kita dapat langsung melatih atau melakukan training dengan menggunakan software Cascade Trainer GUI. Pada saat melakukan training, kita harus menunggu hingga beberapa menit sampai proses training selesai.

Setelah proses training selesai, kita akan mendapatkan file *cascade.xml* yang berfungsi melakukan pendeteksian objek stik game. Selain itu, kita juga dapat melakukan uji performa dari *cascade.xml* nya dengan menggunakan data uji yang telah disediakan. Jika akurasi model sudah memenuhi standar yang diinginkan, maka model tersebut dapat digunakan untuk mendeteksi objek. Proses pendeteksian objek dengan classifier memerlukan pelatihan dan pengolahan data yang cukup besar. Selain itu, hasil pendeteksian juga tergantung pada kemampuan model classifier dalam mempelajari dataset yang tersedia.

3.2. Diagram Blok



- **Start:** Merupakan bagian awal dari diagram blok ini.
- **Cascade Classifier:** Merupakan tahap dimana model cascade classifier digunakan untuk melakukan pendeteksian objek pada gambar yang telah diekstrak fiturnya. Model cascade classifier terdiri dari beberapa lapisan classifier yang di-train secara bertahap, dengan setiap lapisan memiliki tingkat spesifisitas yang lebih tinggi.
- **Acquire Images:** Merupakan proses untuk mengambil gambar-gambar yang akan digunakan sebagai data latih model cascade classifier. Gambar-gambar tersebut dapat diperoleh dari berbagai sumber, seperti kamera atau dataset yang telah tersedia.
- **Preprocessing/Augmentation:** Merupakan proses pengolahan dan pemodifikasi gambar sebelum diberikan ke model cascade classifier. Proses ini dapat meliputi tahap seperti resizing, normalisasi, dan mengubah ke grayscale.
- **Feature Extraction:** Merupakan proses pengambilan fitur-fitur penting dari gambar yang telah diproses. Fitur-fitur tersebut dapat berupa statistik sederhana seperti mean dan standard deviation, atau fitur yang lebih kompleks seperti wavelet transformation.
- **Result:** Merupakan simpul akhir dari diagram blok ini, yang menampilkan hasil dari proses pendeteksian objek menggunakan model cascade classifier.

BAB IV

HASIL DAN PEMBAHASAN

4.1. Hasil

Hasil Output Program :



Output dari deteksi objek dengan cascade classifier adalah daftar objek yang terdeteksi dalam gambar atau video, disertai dengan informasi lokasi (koordinat) masing-masing objek tersebut dalam gambar atau video. Informasi lokasi biasanya dinyatakan dalam bentuk sebuah bounding box yaitu sebuah kotak yang mengelilingi objek tersebut dan menunjukkan letak objek tersebut dalam gambar atau video. Gambar tersebut menunjukkan hasil deteksi objek manusia (man) dalam sebuah gambar, dengan masing-masing objek ditandai dengan sebuah bounding box berwarna biru.

4.2. Pembahasan

1. Pengumpulan Dataset Stik Game

Pengumpulan dataset merupakan tahap penting dalam proses pembuatan model pembelajaran mesin, karena kualitas dataset akan sangat mempengaruhi hasil akhir dari model yang dihasilkan. Oleh karena itu, pengumpulan dataset harus dilakukan dengan cermat dan hati-hati agar dataset yang terkumpul

mewakili dengan baik kelas atau kategori yang akan diklasifikasikan oleh model.

2. Image Labelling

Image labelling proses menandai atau memberikan label kepada objek atau bagian dari objek dalam sebuah gambar. Pada training ini, gambar akan diblock yang menggambarkan apa yang terlihat pada gambar tersebut. Proses ini biasanya dilakukan secara manual dengan gambar atau oleh sebuah algoritma yang dapat mengenali objek dalam gambar.

3. Training Dataset

Training dataset ini digunakan untuk mempelajari sifat-sifat yang terkait dengan masing-masing bentuk stik game, sehingga kemudian dapat memprediksi stik game dari gambar yang belum dikenal sebelumnya dengan tingkat keakuratan yang cukup tinggi. Training dataset juga dapat digunakan untuk melatih model dalam berbagai bidang lain, seperti pemrosesan bahasa alami, analisis data, dan lain-lain. Sekumpulan data yang digunakan sebagai training dataset harus cukup banyak dan merata, agar model yang dihasilkan dapat mewakili sifat-sifat objek atau fenomena yang ingin dipelajari dengan baik.

4. Hasil Training Pengenalan Objek

Pada tahapan ini kita sudah mulai dapat melakukan pengujian hasil training dataset dengan cascade yang dapat membaca objek stik game ataupun objek lain. Proses yang dilakukan adalah dengan menjalankan program di Visual Studio Code dan kemudian program akan membaca objek dengan menggunakan pengujian pada kamera di perangkat laptop.

BAB V

PENUTUP

5.1. Kesimpulan

Program pendeteksian objek stik ini dibuat menggunakan aplikasi Visual Studio Code dan menggunakan *library* OpenCV dan Python. Program pendeteksi ini juga menggunakan Cascade Classifier, yang dapat mendeteksi objek secara video local ataupun secara realtime. Pengumpulan dataset juga merupakan tahap penting dalam proses pembuatan model pembelajaran mesin, karena kualitas dataset akan sangat mempengaruhi hasil akhir dari model yang dihasilkan.

Dalam pendeteksian objek stik ini kita mengumpulkan dataset yang memiliki gambar positif dan juga gambar negatif, dataset positif berisikan berbagai gambar objek yang ingin dideteksi dan dataset negative berisikan gambar acak yang tidak terdapat gambar objek yang ingin dideteksi. Hasil output pada program pendeteksian ini dapat membaca objek stik ketika diarahkan ke kamera.

5.2. Saran

Adapun saran yang dapat penulis sampaikan guna meningkatkan kinerja alat pendeteksi, semakin banyak dan semakin jelas dataset suatu program maka hasil program akan semakin mudah untuk membaca suatu objek yang akan dideteksi.

DAFTAR PUSTAKA

- [1] Muhammad Romzi and B. Kurniawan, “Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma,” *JTIM J. Tek. Inform. Mahakarya*, vol. 03, no. 2, pp. 37–44, 2020.
- [2] G. D. L. Dewi Rosmala, “PEMBANGUNAN WEBSITE CONTENT MONITORING SYSTEM MENGGUNAKAN DIFFLIB PYTHON.” 2012.
- [3] M. Z. Andrekha and Y. Huda, “Deteksi Warna Manggis Menggunakan Pengolahan Citra dengan Opencv Python,” *Voteteknika (Vocational Tek. Elektron. dan Inform.,* vol. 9, no. 4, p. 27, 2021, doi: 10.24036/voteteknika.v9i4.114251.
- [4] N. Gligorijevic, D. Robajac, and O. Nedic, “PERANCANGAN SISTEM INFORMASI PENJUALAN PERUMAHAN MENGGUNAKAN METODE SDLC PADA PT. MANDIRI LAND PROSPEROUS BERBASIS MOBILE,” *J. Teknol. Pelita Bangsa*, vol. 84, no. 10, pp. 1511–1518, 2019, doi: 10.1134/s0320972519100129.
- [5] D. S. R. M. Ninik Sri Lestari1, “Perancangan Aplikasi Pembuatan Kartu Keluarga Berbasis Web Menggunakan Php Dan Mysql,” *Isu Teknol. Stt Mandala*, vol. 15, no. 2, pp. 1–13, 2020.

LAMPIRAN

Source Code file **deteksistik.py** :

```
import cv2

# Mengimport Cascade
stik_cascade = cv2.CascadeClassifier('stik3.xml')

# Membuka kamera untuk mengambil video
cap = cv2.VideoCapture(0)

while True:

    # Membaca frame
    _, img = cap.read()

    # Mengubah frame ke warna grayscale
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    # Mendeteksi objek pada frame
    stik = stik_cascade.detectMultiScale(gray, 1.1, 4)

    # Menandakan objek dengan memberikan kotak disekitar
    for (x, y, w, h) in stik:

        # Menambahkan teks diatas kotak
        cv2.putText(img, 'Joystick', (x, y),
                    cv2.FONT_HERSHEY_SIMPLEX, 1, (120, 111, 67), 2,
cv2.LINE_AA)

        cv2.rectangle(img, (x, y), (x+w, y+h), (255, 0, 0), 2)

    # Menampilkan output
    cv2.imshow('Deteksi Stik', img)
    if cv2.waitKey(1) & 0xff == ord('i'):
        break

# Menjalankan tangkapan video
```

```
cap.release()
```

Source Code file **stik.xml** :

```
<?xml version="1.0"?>
<opencv_storage>
<cascade>
  <stageType>BOOST</stageType>
  <featureType>HAAR</featureType>
  <height>24</height>
  <width>24</width>
  <stageParams>
    <boostType>GAB</boostType>
    <minHitRate>9.9500000476837158e-01</minHitRate>
    <maxFalseAlarm>5.0000000000000000e-01</maxFalseAlarm>
    <weightTrimRate>9.4999999999999996e-01</weightTrimRate>
    <maxDepth>1</maxDepth>
    <maxWeakCount>100</maxWeakCount></stageParams>
  <featureParams>
    <maxCatCount>0</maxCatCount>
    <featSize>1</featSize>
    <mode>BASIC</mode></featureParams>
  <stageNum>15</stageNum>
  <stages>
    <!-- stage 0 -->
    <_>
      <maxWeakCount>4</maxWeakCount>
      <stageThreshold>-1.4073287248611450e+00</stageThreshold>
      <weakClassifiers>
        <_>
          <internalNodes>
            0 -1 48 -2.4474953114986420e-01</internalNodes>
          <leafValues>
            8.7951809167861938e-01 -8.9704477787017822e-
01</leafValues></_>
        <_>
          <internalNodes>
            0 -1 47 -7.6546177268028259e-02</internalNodes>
          <leafValues>
```

```

8.5640263557434082e-01 -7.8664189577102661e-
01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 43 -6.7510366439819336e-02</internalNodes>
    <leafValues>
      7.8525668382644653e-01 -4.6732160449028015e-
01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 32 3.3138439059257507e-02</internalNodes>
    <leafValues>
      -3.9388740062713623e-01 7.4367964267730713e-
01</leafValues></_></weakClassifiers></_>
  <!-- stage 1 -->
  <_>
    <maxWeakCount>6</maxWeakCount>
    <stageThreshold>-1.8598899841308594e+00</stageThreshold>
    <weakClassifiers>
      <_>
        <internalNodes>
          0 -1 36 -4.7533345222473145e-01</internalNodes>
        <leafValues>
          2.6923078298568726e-01 -8.7159532308578491e-
01</leafValues></_>
      <_>
        <internalNodes>
          0 -1 27 -3.8595028221607208e-02</internalNodes>
        <leafValues>
          6.8198657035827637e-01 -6.3343244791030884e-
01</leafValues></_>
      <_>
        <internalNodes>
          0 -1 3 -3.0466372147202492e-02</internalNodes>
        <leafValues>
          5.0696635246276855e-01 -5.2921897172927856e-
01</leafValues></_>
      <_>
        <internalNodes>

```



```

0 -1 88 -3.0670443549752235e-02</internalNodes>
<leafValues>
6.1975973844528198e-01 -3.6511713266372681e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 74 7.4446434155106544e-05</internalNodes>
<leafValues>
2.8224605321884155e-01 -8.0561333894729614e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 73 -1.8334308697376400e-04</internalNodes>
<leafValues>
-8.3324944972991943e-01 2.5722786784172058e-
01</leafValues></_></weakClassifiers></_>
<!-- stage 2 -->
<_>
<maxWeakCount>6</maxWeakCount>
<stageThreshold>-1.7128009796142578e+00</stageThreshold>
<weakClassifiers>
<_>
<internalNodes>
0 -1 54 -1.4897063374519348e-01</internalNodes>
<leafValues>
7.5757575035095215e-01 -8.1255686283111572e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 4 -4.4251881539821625e-02</internalNodes>
<leafValues>
3.8121998310089111e-01 -7.2096031904220581e-
01</leafValues></_>
<_>
<internalNodes>
0 -1 66 5.8138309977948666e-03</internalNodes>
<leafValues>
-5.4832595586776733e-01 3.6920452117919922e-
01</leafValues></_>

```

```

    <_>
    <internalNodes>
      0 -1 60 2.7715428732335567e-03</internalNodes>
    <leafValues>
      -4.2994925379753113e-01 5.2414256334304810e-
01</leafValues></_>
    <_>
    <internalNodes>
      0 -1 94 -9.6590351313352585e-03</internalNodes>
    <leafValues>
      5.4517149925231934e-01 -3.9622193574905396e-
01</leafValues></_>
    <_>
    <internalNodes>
      0 -1 35 -5.3049792768433690e-05</internalNodes>
    <leafValues>
      -6.4857840538024902e-01 2.7768290042877197e-
01</leafValues></_></weakClassifiers></_>
    <!-- stage 3 -->
    <_>
    <maxWeakCount>6</maxWeakCount>
    <stageThreshold>-1.6752988100051880e+00</stageThreshold>
    <weakClassifiers>
    <_>
    <internalNodes>
      0 -1 81 4.2137913405895233e-02</internalNodes>
    <leafValues>
      -8.0342650413513184e-01 1.</leafValues></_>
    <_>
    <internalNodes>
      0 -1 96 6.4326971769332886e-02</internalNodes>
    <leafValues>
      -5.9447592496871948e-01 5.2448743581771851e-
01</leafValues></_>
    <_>
    <internalNodes>
      0 -1 18 -5.8886798797175288e-05</internalNodes>
    <leafValues>
      2.2969134151935577e-01 -6.4696902036666870e-

```

```

01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 23 2.1736769005656242e-02</internalNodes>
    <leafValues>
      1.5208721160888672e-01 -9.0609788894653320e-
01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 24 -8.3361817523837090e-03</internalNodes>
    <leafValues>
      -8.4184777736663818e-01 1.6056229174137115e-
01</leafValues></_>
  <_>
    <internalNodes>
      0 -1 13 1.0505290993023664e-04</internalNodes>
    <leafValues>
      1.8267282843589783e-01 -7.9639542102813721e-
01</leafValues></_></weakClassifiers></_>
  <!-- stage 4 -->
  <_>
    <maxWeakCount>7</maxWeakCount>
    <stageThreshold>-1.9148787260055542e+00</stageThreshold>
    <weakClassifiers>
      <_>
        <internalNodes>
          0 -1 86 8.8948071002960205e-02</internalNodes>
        <leafValues>
          -8.9699572324752808e-01 -1.59999999642372131e-
01</leafValues></_>
      <_>
        <internalNodes>
          0 -1 2 -3.8471810519695282e-02</internalNodes>
        <leafValues>
          2.4630466103553772e-01 -6.1346441507339478e-
01</leafValues></_>
      <_>
        <internalNodes>
          0 -1 14 4.6350891352631152e-05</internalNodes>

```

```
<leafValues>
  2.0900326967239380e-01 -7.3903274536132813e-
01</leafValues></_>
<_>
  <internalNodes>
    0 -1 17 -7.8168101026676595e-05</internalNodes>
    <leafValues>
      -6.2031918764114380e-01 2.7528887987136841e-
01</leafValues></_>
    <_>
      <internalNodes>
        0 -1 34 -1.2906258925795555e-03</internalNodes>
        <leafValues>
          3.1999295949935913e-01 -5.3499686717987061e-
01</leafValues></_>
      <_>
        <internalNodes>
          0 -1 116 3.2901683880481869e-05</internalNodes>
          <leafValues>
            2.7040603756904602e-01 -5.8647930622100830e-
01</leafValues></_>
        <_>
          <internalNodes>
            0 -1 99 -6.2785563059151173e-03</internalNodes>
            <leafValues>
              4.2857033014297485e-01 -4.1118410229682922e-
01</leafValues></_></weakClassifiers></_>
```