

Sistem Deteksi Hama Pada Kolam Budidaya Ikan Berbasis Audio dan Video

Septa Yudha Prasetya¹, I Komang Somawirata², Aryunto Soetedjo³, Radimas Putra M.D.L^{*4}

^{1,2,3,4} Fakultas Teknik, Institut Teknologi Nasional, Malang, Indonesia
septayudhapras@gmail.com¹, kmgsomawirata@yahoo.com², aryunto@gmail.com³,
radimas@lecturer.itn.ac.id^{*4}

Abstrak

Sistem deteksi yang dibangun bertujuan untuk mendeteksi hama pada kolam budidaya ikan. Kolam ikan adalah sumber pendapatan dengan mengelola kolam sebagai wadah atau tempat budidaya ikan konsumsi maupun ikan hias. Akan tetapi pada budidaya kolam ikan mempunyai berbagai permasalahan, salah satunya hama pada kolam ikan terutama hama predator ikan budidaya. hal ini di tunjukan dari banyaknya keluhan dari pembudidaya ikan khususnya pada budidaya ikan air tawar. Berdasarkan permasalahan tersebut peneliti bertujuan untuk merancang sebuah sistem deteksi hama pada kolam budidaya ikan berbasis suara dan video. Berdasarkan hasil pengujian hama, akurasi tingkat kemiripan dengan hama adalah diatas 55%.

Kata Kunci : CNN, *Fast Fourier Transform*, Hama kolam, *Image classifier*, Sistem Deteksi

Abstract

The detection system that was built aims to detect pests in fish farming ponds. Fish pond is a source of income by managing the pond as a container or place for cultivating consumption fish and ornamental fish. However, fish pond cultivation has various problems, one of which is pests in fish ponds, especially pests of cultivated fish predators. This is indicated by the number of complaints from fish cultivators especially in freshwater fish farming. Based on these problems the researcher aims to design a pest detection system in sound and video-based fish farming ponds. Based on the results of pest testing, the accuracy of the level of similarity to pests is above 55%.

Keywords : CNN, *Image classifier*, *Fast Fourier Transform*, *Detection System*, *pond pests*

I. PENDAHULUAN

Di Indonesia kegiatan budidaya perikanan sangat menguntungkan karena di samping memberi keuntungan secara ekonomi pada petani ikan, hal ini juga akan mengurangi import ikan, bahkan ada kemungkinan untuk menembus pasar ekspor. Selain banyak keuntungan, budidaya kolam ikan juga memiliki berbagai permasalahan, terutama hewan yang menjadi hama bagi para petani ikan, Tujuan penelitian ini adalah dihasilkannya metode yang dapat mendeteksi hewan yang menjadi hama bagi ikan budidaya.

Berdasarkan permasalahan yang disampaikan, pada penelitian ini penulis merancang sebuah sistem deteksi hama pada budidaya kolam ikan berbasis suara dan video, beberapa hewan yang di kategorikan menjadi hama bagi para petani ikan adalah Biawak, Berang-Berang, dan masih banyak lagi namun penulis disini hanya mengambil data dari hewan yang telah di sebutkan. Biawak menjadi predator ikan karena ikan merupakan sebuah rantai

makanan bagi biawak, begitupun dengan berang-berang yang menjadikan ikan sebagai makanan utama sehingga hewan satu ini kerap kali meresahkan petani ikan karena porsi makan linsang/berang-berang ini adalah sama dengan berat badan tubuhnya dalam satu kali makan selain itu linsang/berang-berang yang sedang mencari makan datang bersama dengan koloninya. Berdasarkan permasalahan tersebut peneliti bertujuan untuk membuat sistem deteksi hama pada budidaya kolam ikan berbasis suara dan video.

Di sini penulis akan menerapkan ilmu pengolahan citra digital untuk mengolah gambar, untuk metode pengolah gambar yang akan di gunakan oleh penulis akan menggunakan metode CNN (*Convolution neural network*) dan menggunakan metode Fast fourier transform untuk mendeteksi suara.

Pengolahan citra digital merupakan salah satu disiplin ilmu yang mempelajari tentang proses

mengolah citra. Citra yang dimaksud adalah gambar diam (foto) maupun video. Sedangkan digital memiliki arti bahwa pengolahan citra dilakukan secara digital menggunakan komputer[1]. Segmentasi citra merupakan proses memisahkan region atau daerah objek dengan daerah latar belakang agar lebih mudah menganalisa pengenalan objek karena banyak melibatkan persepsi visual[2].

Fast Fourier Transform (FFT) merupakan salah satu algoritma untuk menghitung transformasi fourier diskrit dengan cepat dan efisien. Karena banyak sinyal-sinyal dalam sistem komunikasi yang bersifat kontinu maka di gunakan transformasi *Fourier* yaitu *Fast Fourier Transform*[3].

II. METODOLOGI

2.1. Python

Python merupakan bahasa pemrograman intepretatif multiguna dan juga bahasa pemrograman tingkat tinggi dengan filosofi perancangan yang di fokuskan pada tingkat keterbacaan kode. Python berorientasi pada objek,penafsiran dan interaktif,desain python lebih mudah di baca dan mudah di pahami,karena penggunaan bahasa python sama seperti bahas pemrograman lain dengan *english language* dengan menggunakan rumus dan *syntax* namun lebih sedikit dalam penggunaannya. Python seringkali digunakan untuk keperluan pengembangan perangkat lunak yang dapat di operasikan di *platform* sistem operasi, python adalah salah satu bahasa pemrograman yang *freeware* atay perangkat bebas atau dalam kata lain tidak ada batasan dalam penyalinan atau mendistribusikan.python sudah di lengkapi dengan source code debugger dan profiler. Antarmuka, fungsi sistem GUI (antarmuka pengguna grafis) dan basis datanya[4]–[6].

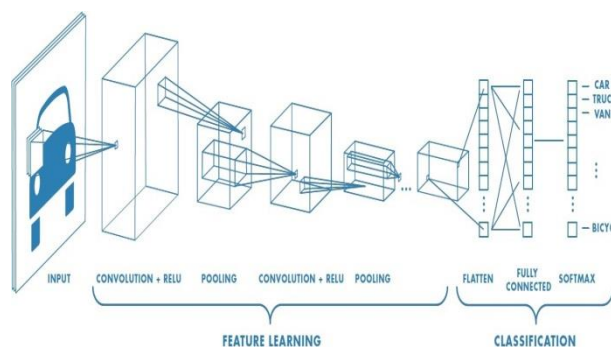
2.2. Deep Learning

Deep Learning merupakan bagian dari *machine learning* yang terdiri dari banyak lapisan (*hidden layer*) dan membentuk tumpukan, lapisan tersebut merupakan metode atau algoritma yang di gunakan untuk melakukan klaisifikasi perintah yang di berikan masukkan sampai menghasilkan *output*. Salah satu metode dari *deep learning* yaitu *Convolutional Neural Network*. Jaringan ini menggunakan masukan berupa gambar, kemudian akan melewati lapisan konvolusi dan diolah berdasarkan *filter* yang telah ditentukan, setiap lapisan ini menghasilkan pola dari beberapa bagian citra yang memudahkan proses

klasifikasi[7].

2.3. Convolution Neural Network

Convolutional Neural Network merupakan salah satu bagian dari *neural network* yang umumnya digunakan dalam pengolahan data *citra*[8][9]. Konvolusi atau yang sering disebut dengan *convolution* merupakan sekumpulan matriks yang berfungsi melakukan *filter*. Seperti ditunjukkan pada Gambar 1. *Convolutional Neural Network* memiliki layer yang berfungsi untuk melakukan *filter* pada setiap proses yang dilakukan. Prosesnya tersebut dinamakan dengan proses *training*. Ada 3 tahapam pada proses *training* yaitu *Convolutional layer*, *pooling layer* dan *fully connected layer*[10]



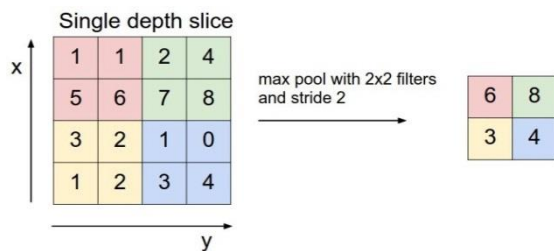
Gambar 1 *Convolution Neural Network* [11]

2.3.1. Convolution Layer

Convolution Layer melakukan operasi konvolusi pada keluaran dari layer sebelumnya. Layer tersebut adalah proses utama yang mendasari CNN[12]. Konvolusi merupakan istilah matematis yang memiliki arti mengaplikasikan sebuah fungsi pada keluaran fungsi lain secara berulang. Dalam pengolahan citra, konvolusi berarti mengaplikasikan sebuah kernel (kotak kuning) pada citra di semua *offset* yang memungkinkan seperti yang ditunjukkan pada Gambar 2. Kotak hijau secara keseluruhan adalah citra yang akan dikonvolusi. Kernel bergerak dari sudut kiri atas ke kanan bawah. Sehingga hasil konvolusi dari citra tersebut dapat dilihat pada gambar disebelah kanannya. Tujuan dilakukannya konvolusi pada data citra yaitu untuk mengekstraksi fitur dari masukan citra. Konvolusi akan menghasilkan transformasi linear dari data masukan sesuai informasi spasial pada data. Bobot pada layer tersebut menspesifikasikan kernel konvolusi yang digunakan, sehingga kernel konvolusi dapat dilatih berdasarkan masukan pada CNN[13].

2.3.2. Pooling Layer

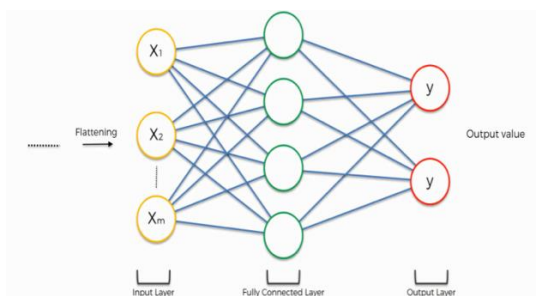
Pooling layer yaitu menerima keluaran dari *convolution layer*, pada *layer* tersebut, ukuran data citra akan direduksi. Prinsipnya *pooling layer* terdiri dari filter yang memiliki ukuran tertentu dan langkah kemudian bergeser keseluruhan area *feature map*. Sebagian metode *pooling* yang sering digunakan yaitu *max pooling*. *Max pooling* yaitu membagi keluaran *convolution layer* menjadi beberapa *grid* kemudian setiap pergeseran *filter* akan mengambil nilai terbesar dari setiap *grid*. Tergantung pada panjang langkahnya, citra digital yang dihasilkan adalah sebagian kecil dari ukuran aslinya yang berguna untuk mengurangi dimensi data, sehingga mengurangi jumlah parameter pada langkah selanjutnya. Pada Gambar 2 menampilkan proses yang ada pada *pooling layer*[11]



Gambar 2 Proses *pooling layer*

2.3.3. Full Connected Layer

Fully connected layer mengambil masukan dari hasil keluaran *pooling layer* yang berupa *feature map*. *Feature map* tersebut masih berbentuk multidimensional array maka hal tersebut akan melakukan *reshape feature map* dan menghasilkan vektor sebanyak *n*-dimensi dimana *n* adalah jumlah kelas keluaran yang harus dipilih program. Misalnya lapisan terdiri dari 500 neuron, maka akan diterapkan *softmax* yang mengembalikan daftar probabilitas terbesar untuk masing-masing 10 label kelas sebagai klasifikasi akhir dari jaringan[11]. Gambar 5 menunjukkan proses yang ada dalam *fully connected layer*[7]



Gambar 3 *Fully Connected Layer*

2.4. Computer Vision

Computer Vision adalah salah satu disiplin ilmu dalam bidang komputer yang mempelajari relasi dan proses transformasi data visual seperti gambar ataupun video menjadi data yang lebih sederhana sehingga dapat diolah oleh komputer untuk menghasilkan suatu keputusan atau informasi baru dari data visual yang diberikan. Berbeda dengan mata manusia yang menangkap data setiap objek secara visual dan dapat mengetahui bentuk dan jarak suatu objek dengan mudah, komputer menangkap tiap data yang ada sebagai susunan angka dalam matrix dua dimensi yang mewakili warna dari data visual yang ada[11]

Human vision sesungguhnya sangat kompleks, manusia melihat objek dengan indra pengelihatannya (mata) lalu onjek citra diteruskan ke otak untuk diinterpretasi sehingga manusia mengerti objek apa yang tampak dalam pandangan matanya. Hasil dari interpretasi ini nantinya yang akan digunakan untuk mengambil suatu keputusan. Inti dari *computer vision* adalah bagaimana sebuah sistem mampu mengenali suatu objek. Untuk itu, *computer vision* diharapkan memiliki kemampuan dengan tingkat tinggi sebagaimana *human visual*. Kemampuan itu diantaranya adalah:

- 1) *Object detection* → Apakah sebuah objek ada pada *scene*.
- 2) *Recognition* → Menempatkan label pada objek.
- 3) *Description* → Menugaskan properti kepada objek
- 4) *3D Inference* → Menafsirkan adegan 3D dari 2D yang dilihat.
- 5) *Interpreting motion* → Menafsirkan gerakan

2.5. Tensorflow

Tensorflow merupakan perpustakaan perangkat lunak yang dikembangkan oleh Tim Google Brain dalam organisasi penelitian mesin cerdas Google, untuk tujuan melakukan pembelajaran mesin dan penelitian jaringan syaraf dalam *Tensorflow* menggabungkan aljabar komputasi teknik pengoptimalan kompilasi, mempermudah penghitungan banyak ekspresi matematis dimana masalahnya adalah waktu yang dibutuhkan untuk melakukan perhitungan. Fitur utamanya meliputi:

- 1) Mendefinisikan, mengoptimalkan, dan menghitung secara efisien ekspresi matematis yang melibatkan array multidimensi (tensors).

- 2) Pemrograman pendukung jaringan syaraf dalam dan teknik pembelajaran mesin.
- 3) Penggunaan GPU yang transparan, mengotomatisasi manajemen dan optimalisasi memori yang sama dan data yang digunakan. Tensorflow bisa menulis kode yang sama dan menjalankannya baik di CPU atau GPU. Lebih khususnya lagi, Tensorflow akan mengetahui bagian perhitungan yang harus dipindahkan ke GPU.

2.6. OpenCV

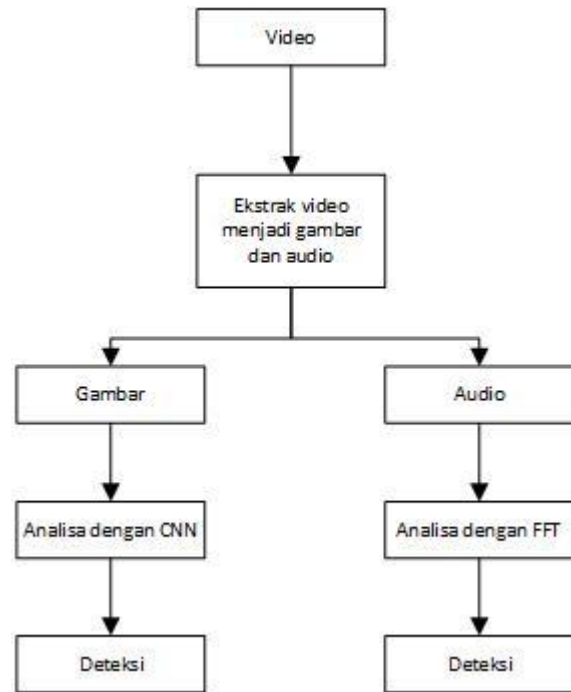
OpenCV adalah sebuah *open source library* untuk *computer vision* yang boleh dipergunakan secara bebas. OpenCV dibuat dengan menggunakan bahasa pemrograman C dan C++ dan dapat dijalankan di berbagai sistem operasi mulai dari Linux, Windows dan Mac OS X. OpenCV memiliki banyak modul yang dapat membantu dalam menyelesaikan bermacam ragam permasalahan terkait komputer Vision. Arsitektur dan manajemen memori yang dimiliki oleh OpenCV yang memungkinkan untuk menjalankan proses dan perhitungan yang kompleks dengan sumberdaya yang terbatas sehingga memberi keleluasaan bagi penggunaanya untuk membangun algoritma pengolahan citra baik itu dengan menggunakan masukkan berupa gambar ataupun video tanpa harus khawatir dengan proses alokasi dan dealokasi memori[11].

2.7. Fast Fourier Transform

Fast Fourier Transform adalah suatu algoritma untuk menghitung transformasi fourier diskrit dengan cepat dan efisien. Karena banyak sinyal-sinyal dalam sistem komunikasi yang bersifat kontinyu, sehingga untuk kasus sinyal kontinyu kita gunakan transformasi fourier. *Fast Fourier Transform* adalah metode yang sangat efisien untuk menghitung koefisien dari Fourier diskrit ke suatu finite sekuen dari data yang kompleks. Karena substansi waktu yang tersimpan lebih dari pada metoda konvensional, fast fourier transform merupakan aplikasi temuan yang penting didalam sejumlah bidang yang berbeda seperti analisis *spectrum*, *speech and optical signal processing*, *design filter digital*.

2.8. Perancangan Sistem

Perancangan sistem deteksi hama berbasis audio dan video dapat di jelaskan pada diagram alir pada gambar 5.



Gambar 4 Diagram blog sistem

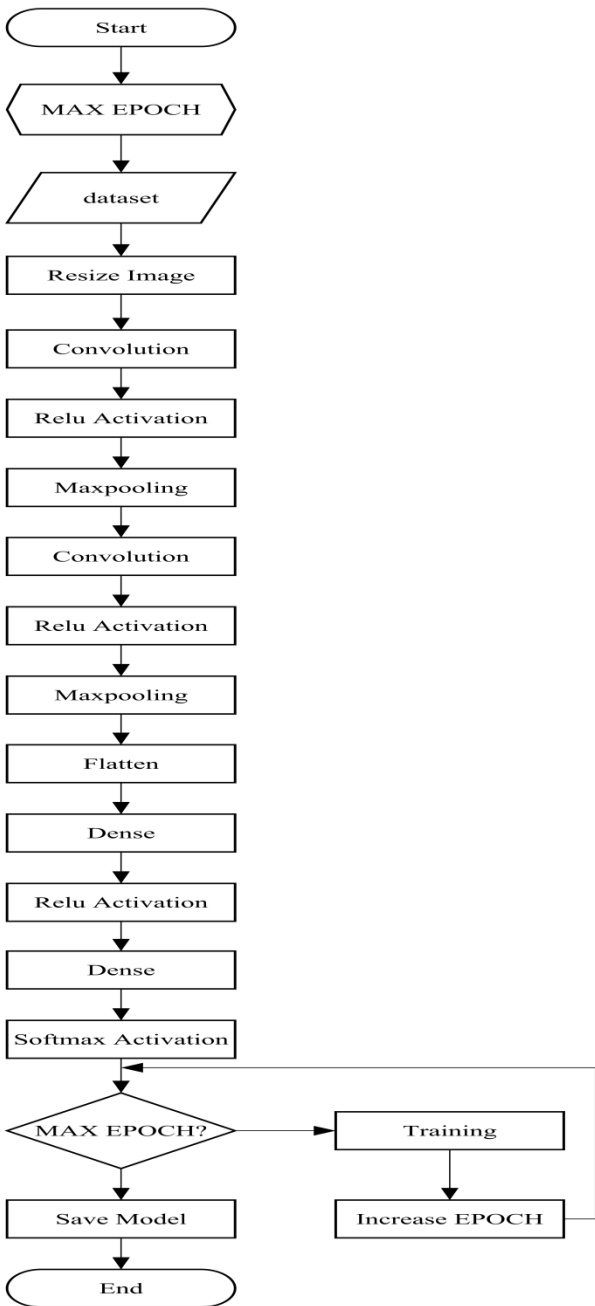
2.9. Prinsip Kerja Sistem

Sistem akan mulai bekerja ketika program python “*test.py*” di jalankan terlebih dahulu menggunakan *command prompt* pada folder program python.”*test.py*” akan mengeksekusi video dengan mengekstraksi gambar dan audio terlebih dahulu, kemudian gambar akan di periksa oleh model (model adalah program yang sudah ada dengan memasukan contoh gambar hama dan gambar bukan hama) dengan ketentuan kemiripan dengan hama di atas 45%. suara hama akan di deteksi menggunakan metode FFT (*Fast Fourier Transform*) dengan cara menentukan ambang batas frekuensi suara hama terlebih dahulu dengan ketentuan frekuensi suara hama di atas 4000Hz. kedua proses ini berjalan bersamaan di karenakan video adalah kumpulan gambar yang berurutan dibaca dengan kecepatan tertentu dengan di tambahkan audio sebagai pengiring gambar tersebut (Diny syarifah, 2015).

2.10. Perancangan Perangkat Lunak

Perancangan perangkat lunak disini berisi tentang penjelasan *flowchart* sistem dan *software* yang mendukung sistem.

2.10.1. Flowchart model

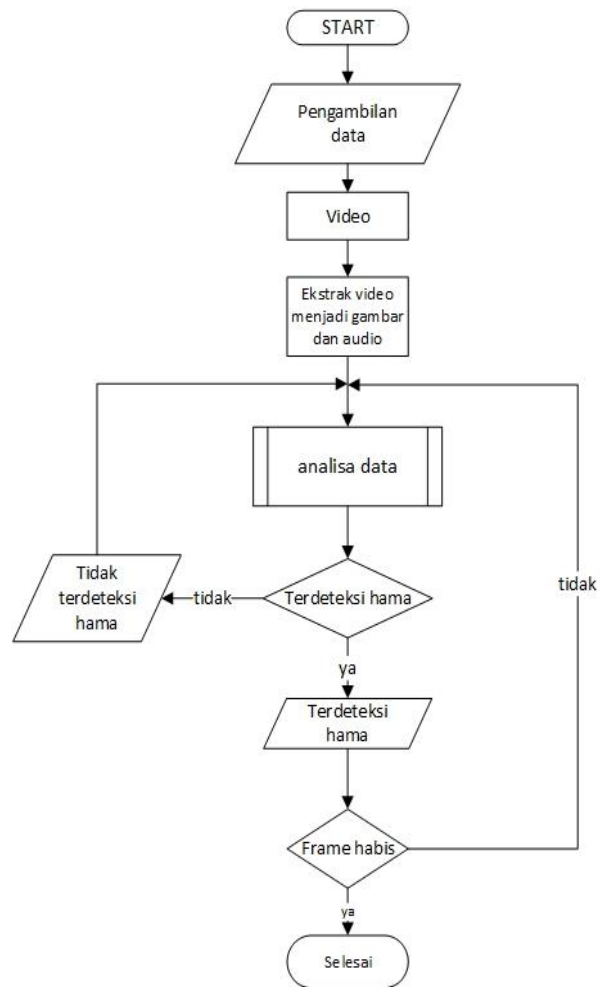


Gambar 5 Flowchart model image

2.10.2. Flowchart Sistem

Flowchart pada gambar 7 merupakan flowchart dengan fungsi yang di jelaskan pada setiap menunya, sistem ini berjalan secara berurutan

1. dimulai dengan pengambilan dan pengumpulan data yang kemudian akan di pilih yang berkualitas baik. data yang baik adalah data yang memiliki sedikit *noise* pada audio dan gambar terlihat jelas dari pixel dan kualitas cahayanya, hal ini di karenakan sistem ini sangat berpengaruh pada kualitas data yang mendeteksi hama berdasarkan prosentase kemiripan dengan data hama pada model.

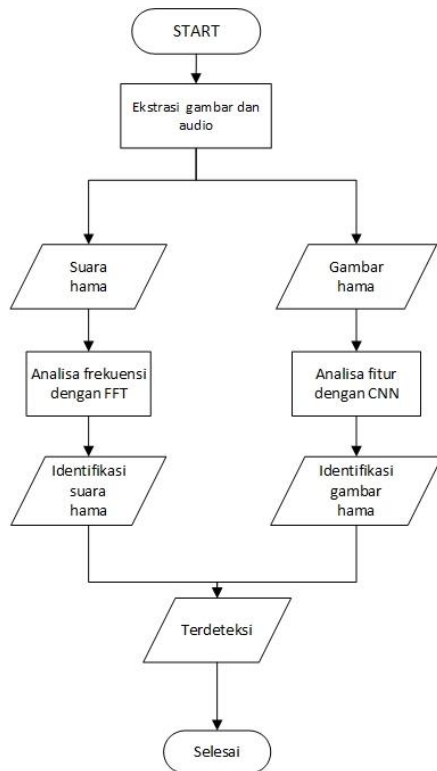


Gambar 6 Flowchart sistem deteksi hama

2. proses kedua pada sistem ini adalah pemilihan video, pemilihan ini bertujuan memastikan bahwa video yang di pilih telah di bedakan menjadi 2, yaitu video hama dan video bukan hama, pemilihan video ini bertujuan agar dapat mengetahui perbedaan antara video hama dan bukan hama ketika program di jalankan.
3. setelah di bedakan menjadi 2 bagian pada proses pemilihan video hama, di sini video mulai di ekstrak menjadi gambar dan suara, proses ini bertujuan karena proses tersebut memiliki metode berbeda dan program yang berbeda untuk mengeksekusi suara dan gambar.
4. proses analisis data hama ini adalah yang menjadi penentu program tersebut berhasil di jalankan atau tidak, karena pada analisis data hama ini ada fungsi yang berisi 2 metode di dalamnya untuk mengeksekusi data hama, penjelasan tentang analisa data hama ada pada gambar 3.4.
5. proses yang terakhir adalah pemeriksaan frame, pemeriksaan frame disini bertujuan

untuk memeriksa program, program tidak akan berhenti menganalisa data hama jika frame belum habis.

2.10.3. Flowchart Analisa data hama



Gambar 7 Flowchart sistem analisa data hama

Tahapan sistem analisa data hama seperti berikut:

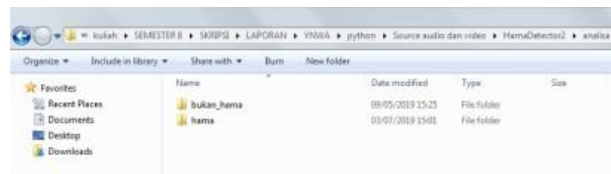
- 1) Tahap awal dari analisa data hama ini adalah mengekstraksi video data hama, dengan tujuan untuk memisahkan antara Gambar dan audio, setelah terpisah maka data tersebut akan di analisa dengan metode yang sesuai.
- 2) Setelah pemisahan data gambar dan suara selesai maka proses analisa gambar dan suara hama akan di jalan sesuai dengan batasan yang telah di tentukan dengan tujuan untuk membedakan suara hama dengan bukan suara hama dan gambar hama dengan bukan gambar hama.
- 3) Data suara hama di analisa dengan menggunakan metode FFT, suara akan terdeteksi sebagai suara hama apabila frekuensinya melebihi dari batas frekuensi yang telah di tentukan begitupun sebaliknya, suara tidak akan terdeteksi jika suara tidak mencapai batasan frekuensi yang sudah di tentukan. ketentuan frekuensi hama jika suara tersebut memiliki frekuensi lebih dari 4000Hz dan tidak dianggap sebagai

suara hama jika di bawah 4000Hz.

- 4) Suara akan terdeteksi sebagai suara hama jika frekuensinya di atas 4000Hz dan tidak di anggap hama jika frekuensinya di bawah 4000Hz.
- 5) Data gambar akan di cocokkan dengan data gambar hama dan gambar bukan hama pada model menggunakan fitur CNN. ketentuan pada fitur CNN untuk data gambar hama adalah tingkat kemiripan dengan prosentase diatas 45%. jika gambar memiliki prosentase di bawah 45% maka di anggap bukan sebagai hama.
- 6) Gambar akan di terdeteksi sebagai hama ketika prosentasi kemiripan hama adalah 45% dan akan terdeteksi bukan sebagai hama jika prosentasenya di bawah 45%.

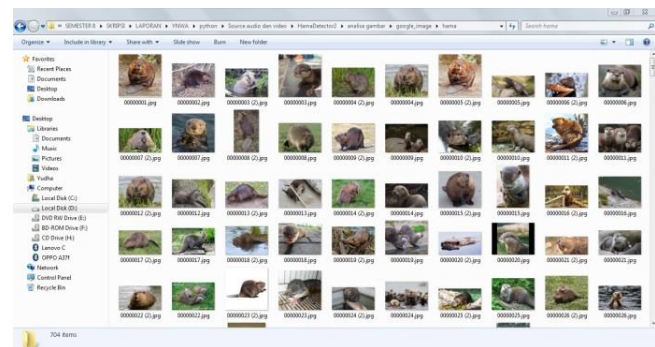
2.11. Training data

Training gambar bertujuan untuk membedakan gambar hama dan bukan, dalam 1 datasheet berisi 2 folder yaitu folder gambar hama dan bukan hama, hal ini bertujuan untuk membedakan gambar hama dan bukan hama. yang akan di gunakan sebagai acuan untuk memastikan gambar yang di uji adalah hama atau bukan.



Gambar 8 folder training

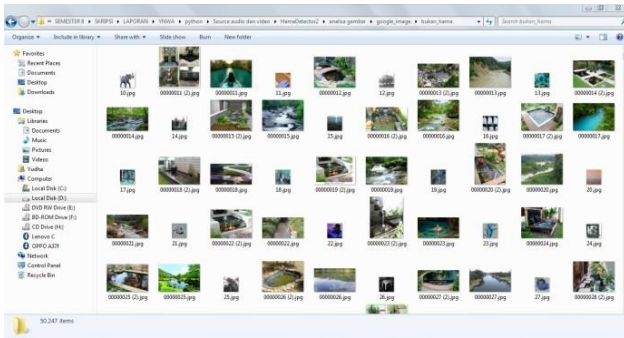
Pada gambar 9, folder training memiliki 2 buah folder yaitu folder hama dan folder bukan hama, pembuatan folder tersebut bertujuan untuk membedakan data hama dan data bukan hama.



Gambar 9 folder training data hama

Pada folder data hama terdapat beberapa gambar hama yaitu gambar berang-berang dan gambar biawak, dari beberapa macam gambar tersebut

program akan mencari nilai gambar yang di anggap sama dengan data hama apabila presentase diatas 40%,

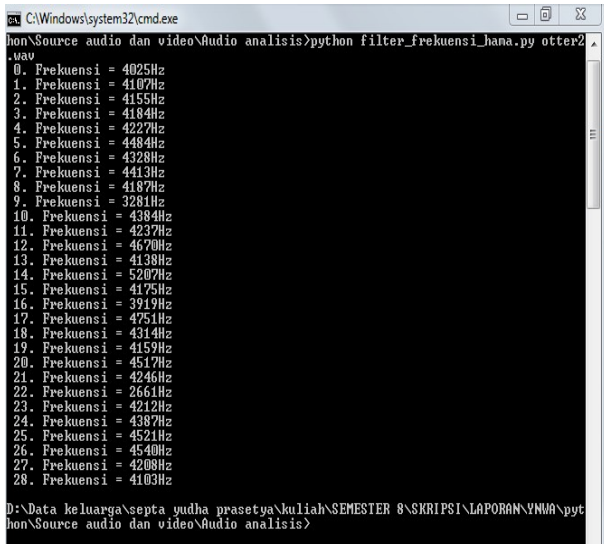


Gambar 10 folder training data bukan hama

Proses akan berlanjut ketika prosentase kemiripan dengan hama berada di bawah 40% dengan memeriksa kemiripan data yang berada di bawah prosentase 40% tersebut dengan data bukan hama, data tersebut terdeteksi secara otomatis sebagai bukan hama dengan presentasi di atas 40%.

2.12. Filter data audio

Proses filter frekuensi pada suara hama bertujuan untuk mengetahui frekuensi hama yang akan di anggap sebagai hama dan bukan hama pada saat uji coba yang di lakukan dengan menggunakan video.



Gambar 11 Proses filter frekuensi suara hama

dari proses training data suara hama di atas dapat di ketahui bahwa suara hama berada pada frekuensi 4000Hz sehingga penentuan suara hama pada program jika frekuensi di atas 4000Hz maka suara tersebut akan di anggap sebagai suara hama dan jika

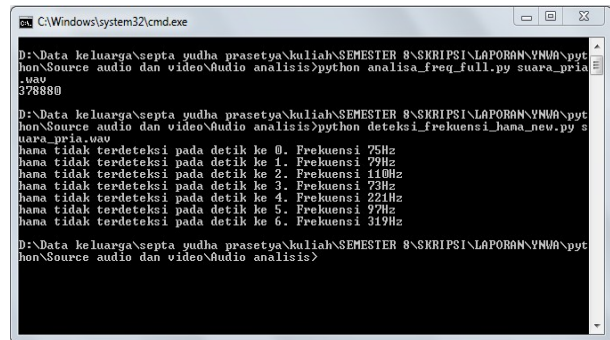
suara tersebut berada di bawah frekuensi tersebut maka bukan suara hama.

III. HASIL DAN PEMBAHASAN

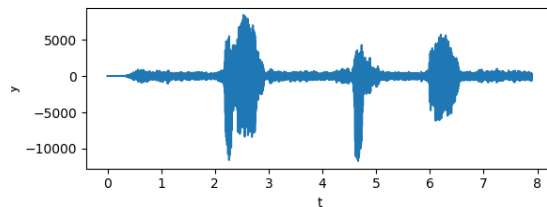
3.1. Pengujian Audio

Pengujian audio disini dilakukan dengan mengambil suara dari beberapa data suara, pengambilan beberapa macam suara disini bertujuan untuk membedakan frekuensi hama dan frekuensi yang ada di sekitarnya.

3.1.1. Suara Manusia Laki-Laki

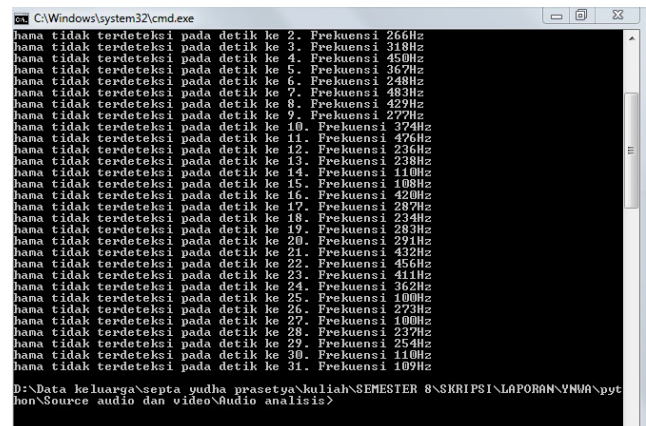


Gambar 12 Frekuensi dan amplitudo suara pria

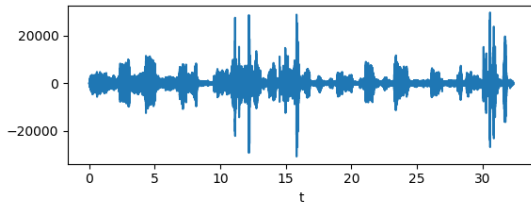


Gambar 13 grafik suara pria

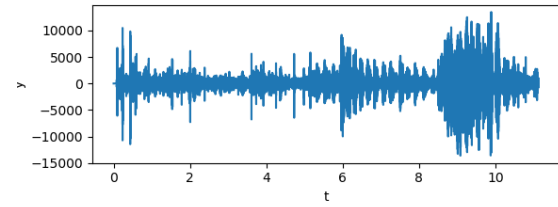
3.1.2. Suara Manusia Perempuan



Gambar 14 Frekuensi dan amplitudo suara wanita



Gambar 15 Grafik suara perempuan



Gambar 19 Grafik suara musik

Dari kedua contoh di atas bahwa frekuensi suara pria berada di bawah 400Hz dan frekuensi suara perempuan ada di bawah 600Hz. kesimpulannya adalah bahwa suara manusia tidak terdeteksi sebagai hama karena suara frekuensi suara manusia diantara 300-600Hz.

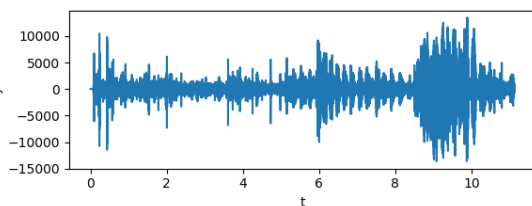
Hasil dari percobaan pada 2 ekor hewan tersebut menunjukkan bahwa frekuensi suara hewan lebih tinggi dari manusia dan memiliki frekuensi yang berbeda, frekuensi suara hewan tidak sama karena tiap hewan memiliki karakter suara yang menjadi ciri khas hewan tersebut sehingga frekuensinya pun pasti berbeda. dari perbedaan suara tersebut memudahkan peneliti untuk memberi batas ketinggian frekuensi untuk suara hama.

3.1.3. Suara Burung Raja Udang

```

C:\Windows\system32\cmd.exe
D:\Data keluarga\septia yudha prasetya\kuliah\SEMESTER 8\SKRIPSI\LAPORAN\YUMA\pyt
hon\Source audio dan video\Audio analisis>python deteksi_frekuensi_hama_new.py t
ing ke k_wau
hama terdeteksi pada detik ke 0. Frekuensi = 37811Hz
hama tidak terdeteksi pada detik ke 1. Frekuensi = 1046Hz
hama tidak terdeteksi pada detik ke 2. Frekuensi = 457Hz
hama terdeteksi pada detik ke 3. Frekuensi = 42234Hz
hama tidak terdeteksi pada detik ke 4. Frekuensi = 556Hz
hama tidak terdeteksi pada detik ke 5. Frekuensi = 3380Hz
hama tidak terdeteksi pada detik ke 6. Frekuensi = 2897Hz
hama tidak terdeteksi pada detik ke 7. Frekuensi = 2910Hz
hama tidak terdeteksi pada detik ke 8. Frekuensi = 2638Hz
hama tidak terdeteksi pada detik ke 9. Frekuensi = 2991Hz
hama tidak terdeteksi pada detik ke 10. Frekuensi = 3352Hz
D:\Data keluarga\septia yudha prasetya\kuliah\SEMESTER 8\SKRIPSI\LAPORAN\YUMA\pyt
hon\Source audio dan video\Audio analisis>
    
```

Gambar 16 Frekuensi dan amplitudo suara burung raja udang



Gambar 17 Grafik suara burung raja udang

3.1.4. Suara Garangan atau Musik

```

C:\Windows\system32\cmd.exe
D:\Data keluarga\septia yudha prasetya\kuliah\SEMESTER 8\SKRIPSI\LAPORAN\YUMA\pyt
hon\Source audio dan video\Audio analisis>python deteksi_frekuensi_hama_new.py g
arangan_wau
hama tidak terdeteksi pada detik ke 0. Frekuensi = 0Hz
hama tidak terdeteksi pada detik ke 1. Frekuensi = 0Hz
hama tidak terdeteksi pada detik ke 2. Frekuensi = 3872Hz
hama terdeteksi pada detik ke 3. Frekuensi = 4112Hz
hama tidak terdeteksi pada detik ke 4. Frekuensi = 3814Hz
hama tidak terdeteksi pada detik ke 5. Frekuensi = 3572Hz
hama tidak terdeteksi pada detik ke 6. Frekuensi = 3566Hz
hama tidak terdeteksi pada detik ke 7. Frekuensi = 3509Hz
hama terdeteksi pada detik ke 8. Frekuensi = 4270Hz
hama tidak terdeteksi pada detik ke 9. Frekuensi = 2982Hz
hama tidak terdeteksi pada detik ke 10. Frekuensi = 3879Hz
D:\Data keluarga\septia yudha prasetya\kuliah\SEMESTER 8\SKRIPSI\LAPORAN\YUMA\pyt
hon\Source audio dan video\Audio analisis>
    
```

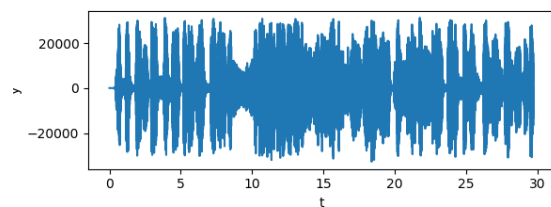
Gambar 18 Frekuensi dan amplitudo suara musik

3.1.4. Suara Berang-Berang

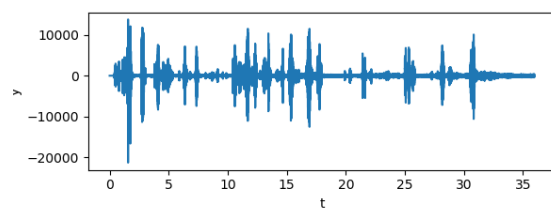
```

C:\Windows\system32\cmd.exe
D:\Data keluarga\septia yudha prasetya\kuliah\SEMESTER 8\SKRIPSI\LAPORAN\YUMA\pyt
hon\Source audio dan video\Audio analisis>python deteksi_frekuensi_hama_new.py o
ter2_wau
hama terdeteksi pada detik ke 0. Frekuensi = 4025Hz
hama terdeteksi pada detik ke 1. Frekuensi = 4107Hz
hama terdeteksi pada detik ke 2. Frekuensi = 4155Hz
hama terdeteksi pada detik ke 3. Frekuensi = 4184Hz
hama terdeteksi pada detik ke 4. Frekuensi = 4227Hz
hama terdeteksi pada detik ke 5. Frekuensi = 4484Hz
hama terdeteksi pada detik ke 6. Frekuensi = 4328Hz
hama terdeteksi pada detik ke 7. Frekuensi = 4413Hz
hama terdeteksi pada detik ke 8. Frekuensi = 4187Hz
hama tidak terdeteksi pada detik ke 9. Frekuensi = 3281Hz
hama terdeteksi pada detik ke 10. Frekuensi = 4384Hz
hama terdeteksi pada detik ke 11. Frekuensi = 4274Hz
hama terdeteksi pada detik ke 12. Frekuensi = 4670Hz
hama terdeteksi pada detik ke 13. Frekuensi = 4138Hz
hama terdeteksi pada detik ke 14. Frekuensi = 5207Hz
hama terdeteksi pada detik ke 15. Frekuensi = 4175Hz
hama tidak terdeteksi pada detik ke 16. Frekuensi = 3919Hz
hama terdeteksi pada detik ke 17. Frekuensi = 4751Hz
hama terdeteksi pada detik ke 18. Frekuensi = 4314Hz
hama terdeteksi pada detik ke 19. Frekuensi = 4159Hz
hama terdeteksi pada detik ke 20. Frekuensi = 4517Hz
hama terdeteksi pada detik ke 21. Frekuensi = 4246Hz
hama tidak terdeteksi pada detik ke 22. Frekuensi = 2661Hz
hama terdeteksi pada detik ke 23. Frekuensi = 4212Hz
hama terdeteksi pada detik ke 24. Frekuensi = 4387Hz
hama terdeteksi pada detik ke 25. Frekuensi = 4521Hz
hama terdeteksi pada detik ke 26. Frekuensi = 4540Hz
hama terdeteksi pada detik ke 27. Frekuensi = 4208Hz
hama terdeteksi pada detik ke 28. Frekuensi = 4103Hz
D:\Data keluarga\septia yudha prasetya\kuliah\SEMESTER 8\SKRIPSI\LAPORAN\YUMA\pyt
hon\Source audio dan video\Audio analisis>
    
```

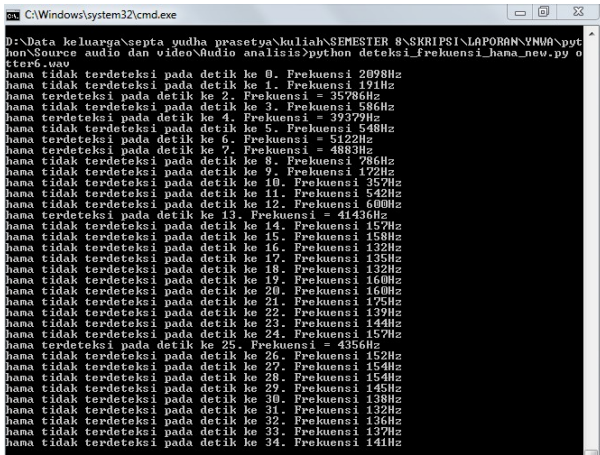
Gambar 20 Frekuensi dan amplitudo suara berang-berang



Gambar 21 Grafik suara Berang-berang



Gambar 22 Grafik suara kedua Berang-berang



Gambar 23 Frekuensi dan amplitudo suara kedua berang-berang

Hasil pengujian pada suara berang-berang di ketahui suara tersebut terdeteksi bahwa suara tersebut sebagai suara hama, karena dari 2 hasil pengujian frekuensi suara yang di dihasilkan sama, yaitu di atas 4000Hz, sehingga program akan menyatakan bahwa suara tersebut adalah suara hama.

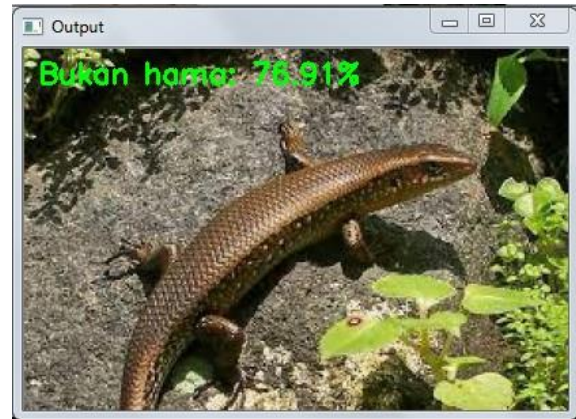
3.2. Pengujian Gambar

Pada proses pengujian gambar, program telah membedakan 2 data gambar yang di beri ketentuan sebagai hama dan bukan hama pada folder datasheet, maka disini program akan langsung mencari kesamaan pada gambar yang ada di datasheet program akan menyatakan bahwa gambar yang di eksekusi tersebut sebagai hama jika prosentase kemiripannya di atas 40% dan akan menyatakan bahwa gambar bukan hama jika prosentasenya di bawah 40%.

3.2.1. Hasil Pengujian Gambar Bukan Hama



Gambar 24 Hasil pengujian gambar musang



Gambar 25 Hasil pengujian gambar kadal



Gambar 26 Hasil pengujian gambar musang



Gambar 27 Hasil pengujian gambar kelinci

Hasil pengujian gambar diatas menunjukkan bahwa gambar bukan termasuk hama karena tingkat kemiripan dengan hama sangat kecil dan tingkat kemiripan dengan bukan hama di atas 50% sehingga program menyimpulkan bahwa gambar tersebut bukan hama.

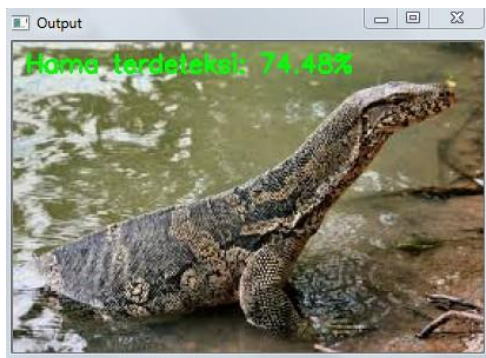
3.3. Hasil Pengujian Hama



Gambar 28 Hasil Pengujian Berang-Berang



Gambar 29 Hasil Pengujian Berang-Berang dengan gambar yang berbeda



Gambar 30 Hasil Pengujian gambar Biawak



Gambar 31 Hasil Pengujian gambar Biawak dengan gambar yang berbeda



Gambar 32 Hasil Pengujian gambar Biawak dengan gambar yang berbeda

Berdasarkan hasil pengujian hama, akurasi tingkat kemiripan dengan hama adalah diatas 55% sehingga proses pendeteksian hama menggunakan sistem yang dibangun berhasil di implementasikan karena pada penelitian ini sudah bisa membedakan antara hama dan bukan hama.

IV. KESIMPULAN

Kesimpulan dari pengujian sistem deteksi hama ini adalah program dapat mendeteksi dan membedakan gambar dan audio sebagai hama dan bukan hama dengan menggunakan metode CNN dan FFT pada program Sistem deteksi hama pada kolam budidaya ikan berbasis audio dan video. Program dapat mengetahui frekuensi suara manusia dan hewan dengan menggunakan metode *fast fourier transform* program dapat menentukan prosentase kemiripan gambar dengan data training dengan menggunakan metode convolution neural network (CNN). Akurasi tingkat kemiripan dengan hama adalah diatas 55% sehingga proses pendeteksian hama menggunakan sistem yang dibangun berhasil di implementasikan.

REFERENSI

- [1] L. Arsy, O. D. Nurhayati, and K. T. Martono, "Aplikasi Pengolahan Citra Digital Meat Detection Dengan Metode Segmentasi K-Mean Clustering Berbasis OpenCV Dan Eclipse," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 2, p. 322, 2016
- [2] M. R. Kumaseh, L. Latumakulita, and N. Nainggolan, "Segmentasi Citra Digital Ikan Menggunakan Metode Thresholding," *J. Ilm. Sains*, vol. 13, no. 1, p. 74, 2013
- [3] P. K. W.P, "Implementasi Fast fourier

- transform dan Divide and Conquer Untuk Pengenalan Huruf Hijaiyah,” 2015.
- [4] G. Varoquaux, L. Buitinck, G. Louppe, O. Grisel, F. Pedregosa, and A. Mueller, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 19, no. 1, pp. 29–33, 2015
- [5] M. Lutz and O. Reilly, “Programming python,” *Comput. Math. with Appl.*, vol. 33, no. 5, p. 132, 1997
- [6] P. Fayyad, U., Piatetsky-Shapiro, G., & Smyth, “Python for Scientific Computing Python Overview,” *Comput. Sci. Eng.*, pp. 10–20, 2007.
- [7] A. Santoso and G. Ariyanto, “Implementasi Deep Learning Berbasis Keras Untuk Pengenalan Wajah,” *J. Emit.*, vol. 18, no. 01, pp. 15–21.
- [8] S. C. B. Lo, S. L. A. Lou, M. V. Chien, and S. K. Mun, “Artificial Convolution Neural Network Techniques and Applications for Lung Nodule Detection,” *IEEE Trans. Med. Imaging*, vol. 14, no. 4, pp. 711–718, 1995
- [9] S. C. B. Lo, H. P. Chan, J. S. Lin, H. Li, M. T. Freedman, and S. K. Mun, “Artificial convolution neural network for medical image pattern recognition,” *Neural Networks*, vol. 8, no. 7–8, pp. 1201–1214, 1995
- [10] W. Huang, Y. Qiao, and X. Tang, “Robust scene text detection with convolution neural network induced MSER trees,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8692 LNCS, no. PART 4, pp. 497–511, 2014
- [11] A. Peryanto, A. Yudhana, and R. Umar, “Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation,” *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 45–51, 2020.
- [12] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, “Image and Video Compression with Neural Networks: A Review,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1683–1698, 2020, doi: 10.1109/TCSVT.2019.2910119.
- [13] H.-J. Yoo, “Deep Convolution Neural Networks in Computer Vision: a Review,” *IEIE Trans. Smart Process. Comput.*, vol. 4, no. 1, pp. 35–43, 2015