PERANCANGAN SISTEM DETEKSI ASAP DAN API MENGGUNAKAN PEMROSESAN CITRA

SKRIPSI

untuk memenuhi salah satu persyaratan

mencapai derajat Sarjana S1

Disusun oleh:

Muhammad Hendri

14524051

Jurusan Teknik Elektro

Fakultas Teknologi Industri

Universitas Islam Indonesia

Yogyakarta

2018

LEMBAR PENGESAHAN

PERANCANGAN SISTEM DETEKSI ASAP DAN API MENGGUNAKAN PEMROSESAN CITRA

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:

Muhammad Hendri
14524051

Yogyakarta, 13 November 2018

Menyetujui,

Pembimbing

Elvira Sukma Wahyuni S.Pd.,M.Eng
155231301

LEMBAR PENGESAHAN

SKRIPSI

PERANCANGAN SISTEM DETEKSI ASAP DAN API MENGGUNAKAN PEMROSESAN CITRA

Dipersiapkan dan disusun oleh:

Muhammad Hendri

14524051

Telah dipertahankan di depan dewan penguji

Pada tanggal: 17 Desember 2018

Susunan dewan penguji

Ketua Penguji : Elvira Sukma Wahyuni S.Pd.,M.Eng, _____

Anggota Penguji 1: Dwi Ana Ratna Wati S.T.,M.Eng , _____

Anggota Penguji 2: Dzata Farahiyah S.T,.M.Sc , _____

Skripsi ini telah diterima sebagai salah satu persyaratan untuk memperoleh gelar Sarjana

Tanggal: 17 Desember 2018

Ketua Program Studi Teknik Elektro

Yusuf Aziz Amrulloh, ST., M.Eng.,Ph.D

045240101

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 17 Desember 2018

Muhammad Hendri

KATA PENGANTAR

AssalamualaikumWr. Wb.

Puji syukur atas rahmat Allah SWT yang telah melancarkan serta memudahkan penulis dalam menyelesaikan tugas akhir ini. Judul dari tugas akhir ini adalah Perancangan Sistem Deteksi Asap dan Api Menggunakan Pemrosesan Citra, harapan penulis dengan adanya tugas akhir ini dapat mempunyai manfaat bagi kalangan mahasiswa dan masyarakat luas. Shalawat serta salam semoga selalu tercurahkan kepada Nabi Muhammad SAW, keluarga serta para sahabat-nya, karena dengan syafa'atnya kita dapat hijrah dari zaman jahiliyah sehingga menuju ke zaman yang terang benderang. Semoga kita dapat menjadi umat-umatnya yang mendapat syafaat Nabi Muhammad SAW di yaumul akhir nanti.

Dengan selesainya tugas akhir ini, penulis ingin berterima kasih dengan segala pihak yang memberikan bantuan, serta bimbingan.

1. Allah SWT, karena dengan rahmat-Nya Tugas Akhir ini dapat diselesaikan dengan lancar.
2. Ibu Elvira Sukma Wahyuni, S.Pd.,M.Eng, selaku dosen pembimbing yang telah banyak memberikan pengarahan dalam proses pengerjaan tugas akhir ini.
3. Seluruh Dosen Jurusan Teknik Elektro, terima kasih atas bimbingan selama menempuh kuliah dari semester pertama hingga akhir di Jurusan Teknik Elektro.
4. Bapak Yusuf Aziz Amrulloh, ST.,M.Eng.,Ph.D Selaku Ketua Jurusan Teknik Elektro Fakultas Teknologi Industri Universitas Islam Indonesia.
5. Orang tua tercinta Ibu Siti Nafsiah dan Bapak Hasan Basri atas segala dukungan yang berupa moral dan materi serta doa yang selalu menyertai sehingga melancarkan segala urusan dalam pelaksanaan tugas akhir ini.
6. Kakak Dian Hastuti dan adik Tri hartati yang telah banyak memberikan bantuan berupa doa dan motivasi untuk menyelesaikan tugas akhir ini.
7. Teman-teman penulis dari Suroto Squad Ahmad, Bayu, Danang, Faiz, Gilang, Gatot, Hardiansyah, Olan, Rahmat, Rendy, Ridho, Riduan, Sabil, Wisnu dan Yoga terimakasih banyak atas keceriaannya, kebersamaan ketika susah maupun senang. Selalu menghibur penulis dalam mengerjakan skripsi ini.
8. Seluruh keluarga besar Teknik Elektro Universitas Islam Indonesia yang telah membantu secara langsung maupun tidak langsung dalam proses pengerjaan Tugas Akhir ini.

9. Dan pihak-pihak lain yang tidak dapat disebutkan satu persatu yang telah membantu secara langsung maupun tidak dalam proses Tugas Akhir ini.

Penulis menyadari bahwa dalam pengerjaan Tugas Akhir ini mempunyai kesalahan dan kekurangan sehingga jauh dalam kata sempurna. Akan tetapi penulis berharap agar dari Tugas Akhir ini dapat menambah wawasan dan ilmu yang bermanfaat bagi penulis maupun orang lain.

Wassalamualaikum Wr. Wb.

Yogyakarta, 10 Agustus 2018

Muhammad Hendri

ARTI LAMBANG DAN SINGKATAN

Daftar Lambang dan Singkatan	Arti
AI	*Artificial Intelligence*
CNN	*Convolutional Neural Network*
PCA	*Principal Component Analysis*
SVM	*Support Vector Machine*
TNKB	Tanda Nomor Kendaraan Bermotor
MLP	*Multi Layer Perceptron*
ANN	*Artificial Neural Network*
PC	*Personal Computer*
GPU	*Graphics Processing Unit*
CPU	*Central Processing Unit*
RGB	*Red, Green, Blue*

ABSTRAK

Penelitian ini dirancang karena kebakaran hutan dan lahan menjadi masalah cukup serius pada Negara-negara yang memiliki iklim tropis, salah satunya Indonesia. Untuk itu dilakukan perancangan sistem deteksi asap dan api menggunakan pemrosesan citra. Perancangan sistem ini menggunakan metode *Convolutional Neural Network* (CNN) sebagai pengolah data citra. Metode ini banyak menghasilkan penelitian yang berguna bagi manusia karena mudah, efektif, efisien, dan akurat. Seluruh *database* penelitian menggunakan citra gambar, dengan data *input* (data *training*) berjumlah 144 data asap dan api sedangkan pada *output* (data uji) berjumlah 20 data asap dan api. Terdapat tiga pengujian model deteksi asap dan api, yaitu berdasarkan jumlah data, jumlah iterasi, dan penggunaan *dropout regularization*. Ketelitian hasil akurasi pengujian berdasarkan nilai rata-rata dari objek yang terdeteksi. Untuk hasil model berdasarkan jumlah data pelatihan didapatkan nilai persentase objek terdeteksi sebesar 100% pada asap dan 54% pada api, jika berdasarkan jumlah iterasi didapatkan nilai persentase objek terdeteksi sebesar 94% pada asap dan 72% pada api, dan apabila menggunakan *dropout regularization* didapatkan nilai persentase objek terdeteksi sebesar 94% pada asap dan 100% api. Pengolahan citra dengan metode *Convolutional Neural Network* pada penelitian ini menghasilkan nilai akurasi yang tinggi sesuai dengan target penelitian yaitu diatas 50%.

Kata kunci : *Convolutional Neural Network*, deteksi objek, *dropout regularization*

DAFTAR ISI

LEMBAR PENGESAHAN i
LEMBAR PENGESAHAN ii
PERNYATAAN iii
KATA PENGANTAR iv
ARTI LAMBANG DAN SINGKATAN vi
ABSTRAK vii
DAFTAR ISI viii
DAFTAR GAMBAR x
DAFTAR TABEL xi
BAB 1 PENDAHULUAN 1
1.1 Latar Belakang Masalah 1
1.2 Rumusan Masalah 2
1.3 Batasan Penelitian 2
1.4 Tujuan Penelitian 2
1.5 Manfaat Penelitian 2
BAB 2 TINJAUAN PUSTAKA 3
2.1 Studi Literatur 3
2.2 Tinjauan Teori 4
2.2.1 Definisi Citra 4
2.2.2 Tipe Citra Digital 4
2.2.3 Pengolahan Citra 4
2.2.4 Convolutional Neural Network 5
2.2.5 *Dropout Regularization* 8
BAB 3 METODOLOGI 9
3.1 Perancangan Perangkat Penelitian 10
3.2 *Database* Dan Pre-*Processing* Data 10

3.3 *Training* Model 10
3.3.1 Konversi *File* XML ke CSV dan *Generate* TFRecord 11
3.3.2 Label Map 12
3.3.3 Konfigurasi Pipeline 12
3.3.4 *Training* Model 12
3.3.5 *Export Graph* 12
3.4 Evaluasi 12
BAB 4 HASIL DAN PEMBAHASAN 13
4.1 Pelabelan Dataset Api dan Asap 13
4.2 Model Hasil Training 14
4.2.1 *Training Steps* 14
4.2.2 *Total Loss* 14
4.2.3 Model 15
4.3 Hasil Deteksi Asap dan Api 16
4.3.1 Hasil Deteksi Berdasarkan Jumlah Data Pelatihan 16
4.3.2 Hasil Deteksi Berdasarkan Jumlah Iterasi Pelatihan 20
4.3.3 Hasil Deteksi Menggunakan *Dropout Regularization* 23
BAB 5 KESIMPULAN DAN SARAN 26
5.1 Kesimpulan 26
5.2 Saran 26
DAFTAR PUSTAKA 27
LAMPIRAN 28

DAFTAR GAMBAR

Gambar 2.1 CNN *layer* 5
Gambar 2.2 Konvolusi Neural Network 6
Gambar 2.3 Proses Konvolusi Neural Network 7
Gambar 2.4 *Average Pooling* 7
Gambar 2.5 Perbandingan Dengan *Dropout* 8
Gambar 3.1 Bagan Alir Metode Penelitian 9
Gambar 3.2 Desain Arsitektur CNN 11
Gambar 4.1 Pelabelan Dataset 13
Gambar 4.2 *Training Step* Pada *Command Window* 14
Gambar 4.3 Grafik *Total Loss* 15
Gambar 4.4 Hasil Model *Training* 15
Gambar 4.5 Gambar Hasil *Training* Model 16
Gambar 4.6 Perbandingan Jumlah Data Pelatihan 20
Gambar 4.7 Perbandingan Jumlah Iterasi Pelatihan 23
Gambar 4.8 Perbandingan Penggunaan *Dropout Regularization* 25

DAFTAR TABEL

Tabel 4.1 Pengujian dengan 112 Data dan 20.000 Iterasi 17
Tabel 4.2 Pengujian dengan 128 Data dan 20.000 Iterasi 18
Tabel 4.3 Pengujian dengan 144 Data dan 20.000 Iterasi 19
Tabel 4.4 Pengujian dengan 60.000 Iterasi dan 144 Data 21
Tabel 4.5 Pengujian dengan 100.000 Iterasi dan 144 Data 22
Tabel 4.6 Pengujian menggunakan *Dropout* dengan 20.000 Iterasi dan 144 Data 24

BAB 1
PENDAHULUAN

1.1 Latar Belakang Masalah

Kebakaran hutan dan lahan menjadi masalah yang cukup serius terutama di negara-negara yang memiliki iklim tropis, salah satunya Indonesia. Faktor terjadinya kebakaran hutan dan lahan adalah unsur kesengajaan dan kelalaian manusia. Faktor kesengajaan antara lain seperti pembakaran hutan untuk membuka lahan baru, lalu fakor kelalaian seperti membakar sampah tanpa pengawasan, dan membuang puntung rokok sembarangan. Jika sudah terjadi kebakaran sering kali api menjadi tidak terkendali dan menjalar mengikuti arah angin. Dampak dari kebakaran hutan dan lahan sangat merugikan manusia, hewan, dan tumbuhan. Seperti menyebabkan pemanasan global, perubahan iklim, tanah longsor, dan banjir.

Teknologi deteksi dini kebakaran hutan dan lahan sangat dibutuhkan dalam tindakan preventif sebelum titik api menjadi lebih luas. Beberapa metode atau teknologi deteksi dini kebakaran telah diterapkan. Salah satunya, sistem deteksi kebakaran dengan mendeteksi asap dan api didalam ruangan menggunakan sensor. Namun sistem tersebut terbatas oleh area deteksi dan tidak memberikan informasi seberapa besar kebakaran itu terjadi. Kekurangan lainnya adalah ketika api kebakaran membesar maka sensor-sensor yang terpasang di gedung dapat terbakar dan rusak[1]. Oleh sebab itu dirancanglah suatu sistem deteksi yang lebih aman untuk mendeteksi kebakaran dari jarak jauh dan dapat memonitoring area yang lebih luas sehingga kerugian atas terjadinya kebakaran dapat diminimalisir.

Teknologi digital mengalami perkembangan yang sangat cepat, dimana hampir setiap aspek dikehidupan manusia membutuhkan teknologi komputasi guna untuk meringankan pekerjaan manusia. Salah satu bidang ilmu penelitian yang berkembang pesat adalah kecerdasan buatan atau yang lebih dikenal dengan sebutan *Artificial Intelligence* (AI). Implementasi dari teknologi AI telah banyak digunakan, baik dalam teknologi *handphone flagship* maupun dalam dunia robotika. Salah satu cabang ilmu pengetahuan dari *Artificial Intelligence* adalah *computer vision* yang mempelajari disiplin ilmu tentang bagaimana komputer dapat mengenali objek yang diamati.

Teknologi *computer vision* merupakan kombinasi antara pengolahan citra (*image processing)* dan pengenalan pola (*pattern recognition*). Dimana pengolahan citra merupakan bidang yang berhubungan dengan transformasi citra dengan tujuan mendapatkan kualitas citra yang lebih baik. Sedangkan pengenalan pola merupakan bidang yang berhubungan dengan proses identifikasi objek pada citra dengan tujuan untuk mengekstrak informasi yang terdapat pada citra.

Pada penelitian ini mencoba menerapkan metode deteksi berbasis citra, yaitu *Convolutional Neural Network.* CNN telah banyak digunakan dalam penelitian pengolahan citra (*image processing*). Metode ini memiliki tingkat akurasi yang tinggi dalam mendeteksi objek maupun *image classification*. Sehingga pada penelitian ini dikembangkan metode deteksi dini kebakaran berbasis citra menggunakan algoritma CNN.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, maka dibuat rumusan masalah sebagai berikut :

Bagaimana merancang dan mengimplementasikan sistem deteksi asap dan api menggunakan pemrosesan citra ?

1.3 Batasan Penelitian

1. Penelitian ini menggunakan gambar sebagai *input* dan *output*
2. *Database* yang digunakan pada penelitan ini merupakan api dan asap yang besar dan tebal

1.4 Tujuan Penelitian

Berikut adalah tujuan dari penelitian yang ingin dicapai sebagai berikut :

Terciptanya suatu sistem deteksi dini asap dan api dengan teknik pemrosesan citra untuk memonitoring dan mencegah terjadinya kebakaran.

1.5 Manfaat Penelitian

Berikut adalah manfaat dalam penelitian yang ingin dicapai sebagi berikut :

1. Meminimalisir bencana kebakaran yang terjadi di Indonesia
2. Mengurangi korban jiwa saat terjadi kebakaran
3. Dapat melakukan evakuasi dini jika terjadi kebakaran
4. Mencegah kerusakan pada lingkungan sekitar
5. Memonitoring ruangan maupun lahan disekitar terjadinya kebakaran

BAB 2
TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian tentang *image processing* telah banyak dilakukan. Penelitian dari Gao xu tentang mendeteksi asap menggunakan *synthetic smoke images.* Hal pertama yang dilakukan penelitian ini adalah membangun sebuah sintesis pipa dan mengatur berbagai kondisi untuk simulasi asap serta merendering secara acak agar menghasilkan sintesis gambar asap yang beragam. Langkah kedua yaitu membagi dataset menjadi beberapa sumber *(synthetic real smoke and non-smoke). Multi-label deep architecture* dibangun berdasarkan adaptasi domain dan berfungsi untuk mengekstrak fitur *invariant* domain dari sintesis gambar asap. Dalam pengujiannya non-asap memiliki gangguan yang kuat terhadap pengenalan asap sehingga mudah menyebabkan *alarm* palsu[2].

Penelitian lain tentang *image processing* oleh Dika abadianto ekorianto tentang perancangan sistem pengenalan wajah untuk mendukung *smart home system.* Pada penelitian ini sistem dirancang dengan menggunakan *image processing* dan menggunakan metode *principal component analysis (PCA).* Sistem pengenalan wajah dirancang untuk keperluan *smart home system* sehingga keamanan rumah lebih terjamin. Pada pengujian sistem pengenal wajah tersebut didapat nilai akurasi sebesar 80% dengan batasan berupa jumlah orang yang tidak lebih dari empat, warna baju yang mesti berbeda dari warna kulit, dan tidak menggunakan latar belakang bewarna kulit, serta jarak antar kamera dan objek ≤ 240 cm[3].

Penelitian lain juga dilakukan oleh Dileep k tentang *a video-based smoke detection using smoke flow pattern ann spatial-temporal energy analyses for alarm systems.* Pada penelitian ini memiliki tiga parameter penting dalam merancang sistem deteksi asap yaitu perilaku difusi, warna, dan kekaburan (*blur).* Hal pertama yang dilakukan penelitian ini adalah menganalis warna, lalu mengekstrak fitur dengan menggunakan metode *Gabor Filtering* dan *Spatial Temporal Energy Analysis* agar mendapatkan vektor fitur. Tahap terakhir penilitian ini yaitu mengklasifikasikan jenis asap dengan *Support Vector Machine (SVM).* Pada pengujian sistem deteksi asap tesebut menggunakan *software Matlab 2015* dengan *frame resolution* sebesar 320 x 240 piksel[4].

Penelitian mengenai *deep learning* untuk deteksi tanda nomor kendaraan bermotor dengan algortima *convolutional neural network* menggunakan *python* dan TensorFlow pada tahun 2018 yang dilakukan oleh Imam Taufik. Pada penelitian ini menggunakan algoritma *convolutional neural network* untuk mendeteksi Tanda Nomor Kendaraan Bermotor (TNKB). Penelitian ini bertujuan untuk mendapatkan sebuah model CNN dengan tingkat akurasi tinggi ketika mendeteksi objek dari plat nomor kendaraan bermotor[5].

2.2 Tinjauan Teori

2.2.1 Definisi Citra

Citra (*image*) merupakan suatu gambar dua dimensi yang terbentuk dari susunan piksel. Pada umumnya, citra dibentuk oleh kotak-kotak persegi empat yang teratur sehingga jarak horizontal dan vertikal memiliki nilai piksel yang sama pada seluruh bagian citra. Citra dapat dikategorikan menjadi dua bagian yaitu citra diam (*still image*) dan citra bergerak (*moving image*). Citra bergerak merupakan gabungan dari citra diam yang ditampilkan secara beruntun, sehingga memberi kesan pada mata bahwa citra tersebut bergerak.

2.2.2 Tipe Citra Digital

Citra digital dapat dikategorikan sebagai berikut:

1. Citra Biner

   Citra biner adalah citra yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam dan putih. Hanya dibutuhkan 1 bit untuk mewakili nilai setiap piksel dari citra biner.
2. Citra *Greyscale*

   Citra *greyscale* adalah citra yang disetiap pikselnya mengandung satu *layer* dimana nilai intesitasnya berada pada nilai 0 (hitam) – 255 (putih).
3. Citra Warna

   Citra warna(RGB) adalah citra yang memiliki informasi warna pada setiap pikselnya

2.2.3 Pengolahan Citra

Pengolahan citra merupakan suatu proses pengolahan data menggunakan komputer sehingga citra tersebut dapat menjadi sebuah citra yang memiliki kualitas baik. Tujuan dari pengolahan citra ini adalah untuk memperbaiki kualitas suatu citra agar dapat diinterpretasi dengan manusia maupun komputer. Pengekstrasian informasi citra membutuhkan suatu teknik agar dapat mengelompokkan (*clustering)* piksel-piksel yang ada guna menyederhanakan proses pengenalan unsur-unsur spasial yang ada didalamnya. Pengklasifikasian citra merupakan suatu penyusunan, pengurutan maupun pengelompokkan semua piksel ke dalam beberapa kelompok dengan berdasarkan kriteria objek. Teknik klasifikasi dalam pemrosesan citra dibagi menjadi dua yaitu: klasifikasi terawasi (*supervised classification)* dan klasifikasi tak terawasi *(unsupervised classification).* Pada klasifikasi terawasi digunakan untuk mendefenisikan kelas-kelas oleh *user*. Kelas-kelas yang dimaksud berupa sampel-sampel yang diasumsikan memiliki sifat homogen.

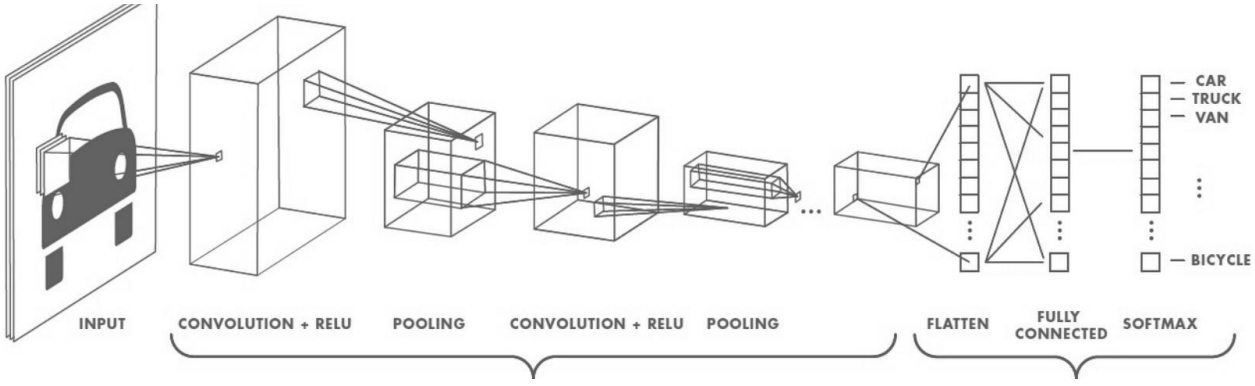
Sedangkan klasifikasi tak terawasi digunakan untuk mengklasifikasikan citra berdasarkan aspek statistik semata[6].

Ekstraksi fitur citra secara umum dibagi menjadi dua karakteristik piksel yaitu similaritas dan perbedaan kedekatan nilai-nilai piksel. Salah satu metode ekstrasi citra yaitu *thresholding.* Metode ini memiliki kriteria similaritas yang didasarkan pada jangkauan nilai-nilai *grey* yang termasuk pada fitur yang bersesuaian. Terdapat berbagai macam teknik dalam pemrosesan citra diantaranya *image differencing technique* yang merupakan teknik paling sederhana, yaitu dengan cara membandingkan selisih citra baru dan citra lama. *Principle Component Analysis* yaitu teknik yang digunakan untuk menyederhanakan suatu data, dengan cara mentransformasi data secara linear sehingga terbentuk sistem koordinat baru dengan varian maksimum. *Convolutional neural network* terdiri dari *Neuron, Convolution layer, Activation(*Relu), *pooling layer, fully connected layer* sehingga semua *layer* tersebut akan dikonvolusikan secara linear.

2.2.4 Convolutional Neural Network

Convolutional neural network (CNN) merupakan pengembangan dari *Multi Layer Perceptron* (MLP) yang didesain untuk mengolah data dua dimensi dalam bentuk citra. *Convolutional Neural Network* merupakan *neural network* yang didalamnya terdapat konvolusi minimal pada salah satu lapisannya, serta menjadi *special case* dari *Artificial Neural Network* (ANN) yang saat ini diklaim sebagai model terbaik untuk memecahkan permasalahan pada *object recognition* maupun *object detection.*

Convolutional neural network termasuk dalam jenis *Deep Neural Network* karena memiliki kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Arsitektur dalam *Convolutional neural network* dapat di *training* dan terdiri dari beberapa tahap. *Input* dan *output* dari masing-masing tahap adalah beberapa *array* yang disebut *feature map* atau map fitur. Terdapat empat *layer* utama dalam algoritma *Convolutional neural network* yaitu *convolutional layer, pooling layer, activation layer,* dan *fully connected layer*[7].



Gambar 2.1 CNN *layer*

2.2.4.1 *Convolutional Layer*

Convolutional layer merupakan *layer* pertama yang menerima *input* dari gambar langsung pada arsitektur jaringan. *Convolutional layer* melakukan operasi konvolusi pada *output* dari *layer* sebelumnya. Tujuan utama pada konvolusi data citra adalah untuk mengekstrasi fitur dari data citra *input*. Pada umumnya operasi konvolusi menggunakan persamaan sebagai berikut.

$$s(t) = (x * w)(t) \tag{2.1}$$

Keterangan:

s(t) : fungsi hasil proses konvolusi

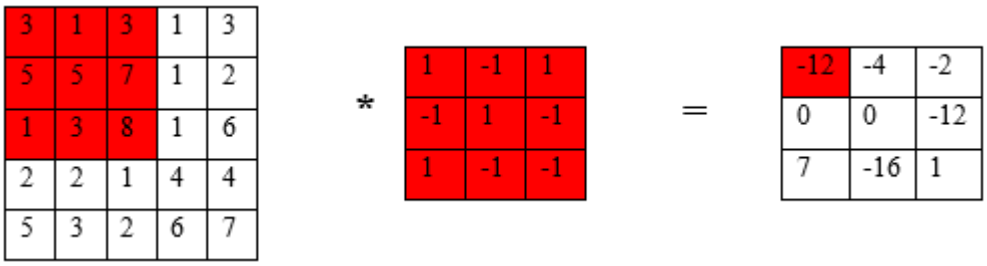
x : *input*

w : bobot (kernel)

Fungsi s(t) memberikan *output* tunggal berupa *feature map*. Argumen pertama berupa *input* yang merupakan (x) dan argument kedua yaitu (w) sebagai kernel atau *filter*. Jika melihat input sebagai citra dua dimensi, maka (t) diasumsikan sebagai sebuah piksel dan menggantinya dengan i dan j. Jika operasi konvolusi lebih dari satu dimensi dapat menggunakan persamaan berikut.

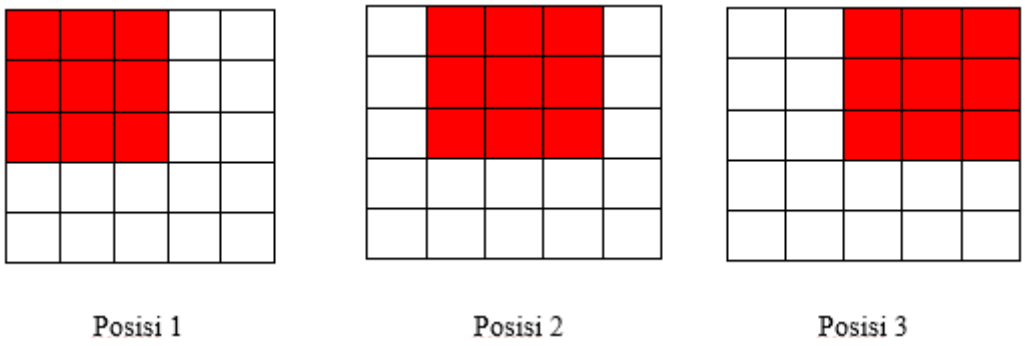
$$S_{(i,j)} = (K * I)_{(i,j)} = \sum\sum I_{(i-m,j-n)}\, K_{(m,n)} \tag{2.2}$$

Dari persamaan diatas merupakan perhitungan operasi konvolusi dengan i dan j sebagai piksel dari sebuah citra. (K) merupakan sebuah kernel atau *filter* yang dapat dibalik relatif terhadap *input* serta (I) sebagai *input*[8]. Operasi konvolusi dapat dilihat sebagai perkalian matriks antara citra *input* dan filter dimana *output*nya dapat dihitung dengan *dot product*. Untuk dapat lebih memahami dari prinsip kerja konvolusi, peneliti akan menggunakan sampel dengan *input* matriks 5x5 dikarenakan keterbatasan penulisan dengan ukuran 300x300 serta menggunakan *filter* untuk operasi deteksi tepi dengan ukuran 3x3.



Gambar 2.2 Konvolusi Neural Network

Gambar 2.2 adalah proses perhitungan konvolusi dengan menggunakan ukuran filter 3x3, dan menggunakan pergeseran *filter* terhadap matriks *input* berjumlah satu. Jika divisualisasikan sebagai berikut:



Gambar 2.3 Proses Konvolusi Neural Network

Gambar 2.3 merupakan perhitungan pada proses konvolusi dimana sebuah *filter* dengan ukuran 3x3 yang diawali pada sisi bagian kiri proses ini disebut dengan *sliding window*. Contoh proses perhitungan proses konvolusi dari posisi 1 hingga ke posisi 3 sebagai berikut :

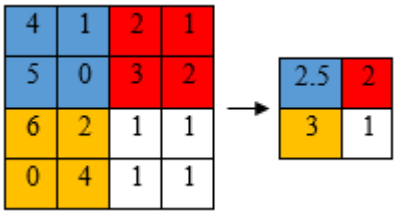
a. Posisi 1 = (3x1)+(5x(-1))+(1x1)+(1x(-1))+(5x1)+(3x(-1))+(3x1)+(7x(-1))+(8x-(1)) = -12
b. Posisi 2 = (1x1)+(5x(-1))+(3x1)+(3x(-1))+(7x1)+(8x(-1))+(1x1)+(1x(-1))+(1x1) = -4
c. Posisi 3 = (3x1)+(7x(-1))+(8x1)+(1x(-1))+(1x1)+(1x(-1))+(3x1)+(2x(-1)+(6x(-1)) = -2

Terdapat dua parameter untuk memodifikasi *layer* yaitu.

1. *Stride* adalah parameter yang menentukan berapa jumlah pergeseran *filter*, jika nilai *stride* satu maka *feature map* akan bergeser sebanyak satu piksel secara horizontal dan vertikal. Semakin kecil nilai *stride* maka hasil akan semakin detail serta membutuhkan komputasi lebih jika dibandingkan dengan nilai stride yang besar.
2. *Padding* merupakan parameter yang menentukan jumlah piksel (bernilai nol) yang akan ditambahkan pada tiap sisi dari *input*. Tujuan dilakukan *padding* untuk mengatur nilai *output* agar sama dengan *input* atau tidak terlalu berkurang drastis sehingga dapat dilakukan ekstrasi *feature* lebih mendalam.

2.2.4.2 *Pooling Layer*

Pooling atau *subsampling* merupakan pengurangan ukuran matriks yang biasanya dilakukan setelah operasi *convolutional layer*. Terdapat dua macam *pooling* yang sering digunakan yaitu *average pooling* dan *max pooling*. Dalam *average pooling* nilai yang diambil adalah nilai rata-rata dari *input* awal matriks.



Gambar 2.4 *Average Pooling*

Output dari *pooling layer* yaitu berupa matriks yang lebih kecil dibandingkan dengan matriks awal. Proses konvolusi dan *pooling* dilakukan beberapa kali sehingga mendapatkan hasil yang diinginkan.

2.2.4.3 *Activation Function*

Activation function merupakan sebuah *node* yang ditambahkan diakhir keluaran dari setiap jaringan syaraf. Pada arsitektur CNN, fungsi aktivasi digunakan pada perhitungan akhir keluaran *feature map* atau setelah proses konvolusi maupun *pooling layer*. Terdapat banyak fungsi aktivasi yang sering digunakan dalam *neural network* yaitu sigmoid, tanh, ReLu(*Rectified Linear*), parameter ReLu, dan leaky ReLu.

2.2.4.4 *Fully Connected Layer*

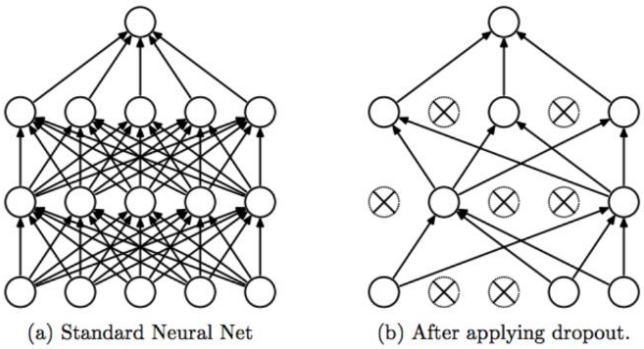
Sebelum masuk ke *fully connected layer*, keluaran dari *feature map* masih berbentuk multidimensional *array*. Sehingga perlu melakukan *flatten* atau *reshape feature map* menjadi sebuah *vector* agar dapat digunakan sebagai *input fully connected layer*.

Pada lapisan *fully connected layer*, semua *neuron* aktivasi dari lapisan sebelumnya terhubung pada *neuron* dilapisan selanjutnya sehingga seperti jaringan syaraf tiruan biasa. Pada lapisan ini pula menghasilkan *output* berupa klasifikasi citra yang diinginkan.

2.2.5 *Dropout Regularization*

Regularization merupakan teknik yang digunakan untuk mengurangi *overfitting* atau *noise.* Yaitu kondisi dimana sistem mampu belajar dengan baik dengan data pelatihan, namun tidak dapat menggeneralisasi dengan data uji.

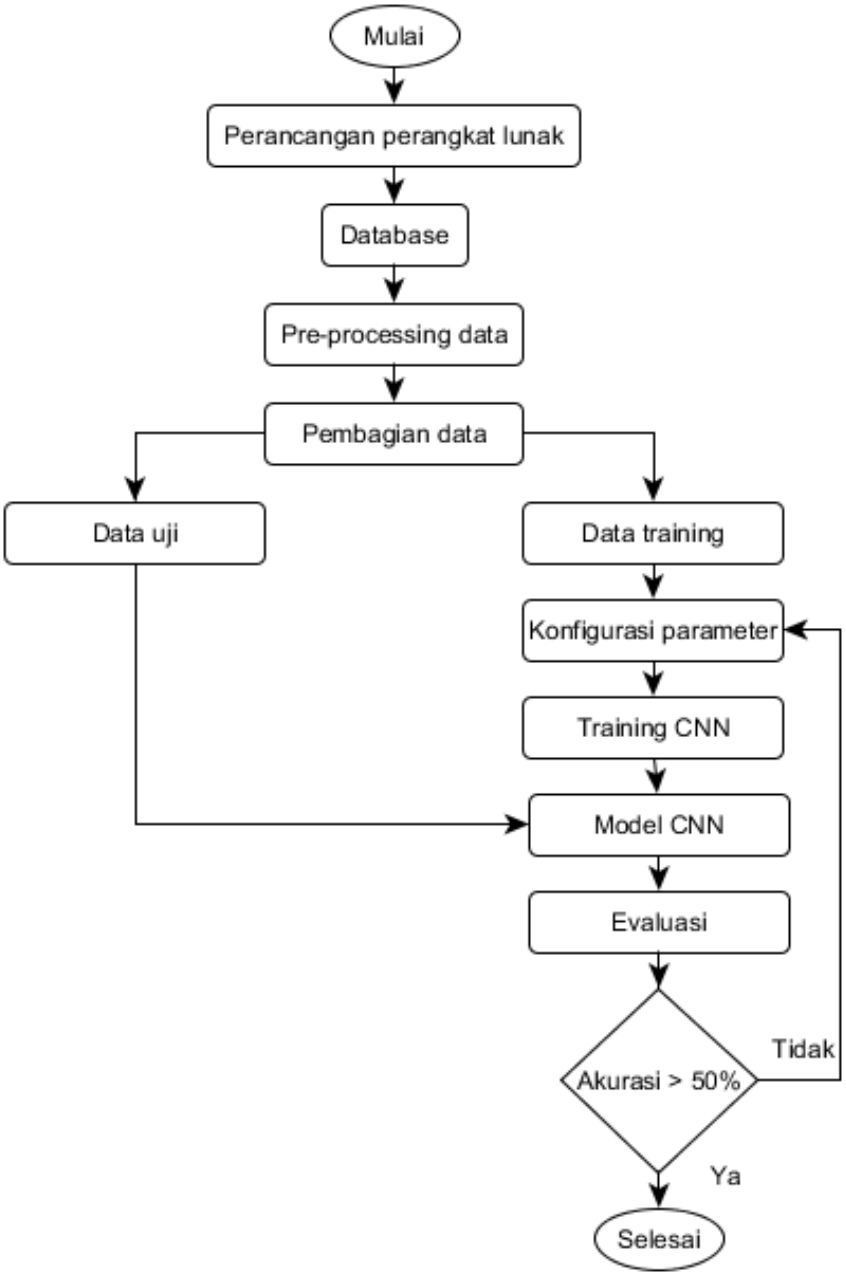
Dropout merupakan teknik yang digunakan untuk mencegah *overfitting* serta mempercepat proses pelatihan. *Dropout* menghilangkan *neuron* yang berupa *hidden* maupun *layer* yang terlihat pada jaringan. *Neuron* yang hilang akan dipilih secara acak oleh sistem dengan probabilitas dari nol hingga satu. Berikut gambaran dari proses saat adanya *dropout*[9].



Gambar 2.5 Perbandingan Dengan *Dropout*

BAB 3
METODOLOGI

Pada Bab ini menjelaskan metode yang digunakan untuk menyelesaikan rumusan masalah yang diteliti. Metode yang digunakan dalam mendeteksi asap dan api menggunakan algoritma *convolutional neural network* dapat dilihat pada bagan alir gambar 3.1.



Gambar 3.1 Bagan Alir Metode Penelitian

3.1 Perancangan Perangkat Penelitian

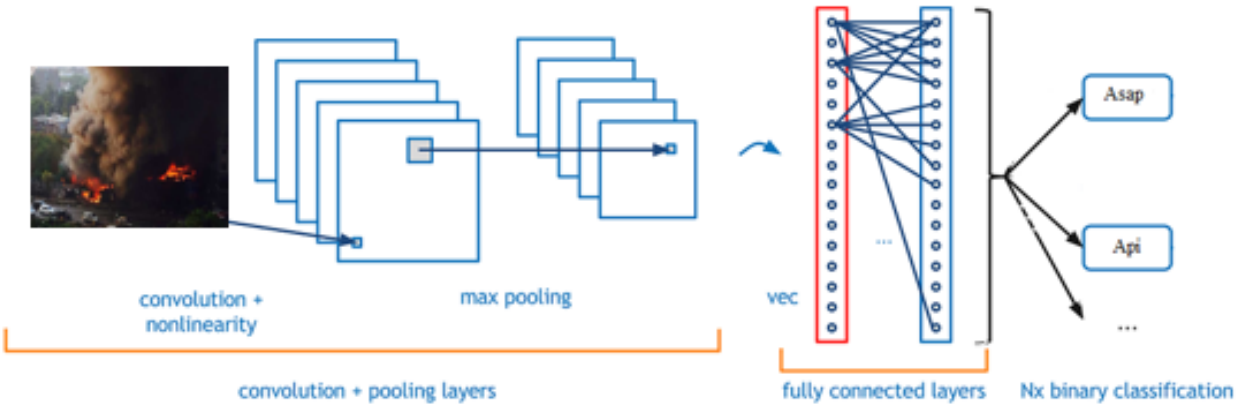
Sistem deteksi asap dan api ini memerlukan perangkat keras berupa *personal computer* (PC). Sistem ini mendeteksi objek secara gambar. *Personal computer* (PC) digunakan untuk memonitor serta sebagai pengendali sistem. Perangkat lunak yang digunakan dalam penelitian ini yaitu *Python-3, Visual Studio* 2015, *Cuda*-9 *Nvdia.* Bahasa program yang digunakan pada *Python* adalah bahasa *C, Visual studio* berguna sebagai editor program, dan *Cuda* berfungsi untuk menjalankan program menggunakan *Graphics Processing Unit* (GPU) agar kemampuan komputasi menjadi lebih cepat dibandingkan *Central Processing Unit (*CPU*)*. Pada penelitian ini menggunakan *framework* dari Google yaitu TensorFlow, dimana *framework* ini banyak digunakan untuk keperluan *deep learning, machine learning,* maupun *artificial intelligence.*

3.2 *Database* Dan Pre-*Processing* Data

Langkah awal dalam mendeteksi objek adalah mempersiapkan data berupa citra asap dan api dalam format JPG maupun JPEG. Citra yang dijadikan *database* pada penellitian ini berasal dari gambar Google yang di unduh secara acak. *Input* dari rancangan sistem yang digunakan peneliti berupa gambar dengan resolusi yang berbeda-beda, namun pada saat *pre-processing* ukuran data *input* diseragamkan menjadi 300x300 piksel. Jumlah total data yang digunakan peneliti sebanyak 164 gambar dari asap dan api. Data tersebut dibagi menjadi dua bagian yaitu data *training* dan data uji, dimana data *training* menggunakan 144 gambar dan data uji 20 gambar (asap dan api). Pada data *training* terdapat 20 gambar api, 65 gambar asap, dan 59 gambar campuran. Sedangkan pada data uji terdapat 1 gambar api, 9 gambar asap, dan 10 gambar campuran. Sebelum melakukan *training* model seluruh data *training* diklasifikasikan sesuai dengan objek yang ingin dideteksi yaitu api dan asap yang bertujuan untuk memudahkan komputer mempelajari objek deteksi.

3.3 *Training* Model

Dalam proses *training* model, algoritma *Convolutional Neural Network* membutuhkan desain arsitektur. Desain arsitektur ini terdiri dari beberapa tahapan yaitu *input Neuron,* konvolusi *layer, Activation(*Relu), *pooling layer, fully connected layer,* klasifikasi dan deteksi objek. Berikut tahapan dari arsitektur jaringan *Convolutional Neural Network*:



Gambar 3.2 Desain Arsitektur CNN

Pada gambar 3.2 merupakan desain aristektur CNN dimana *input* gambar akan beresolusi secara acak dengan warna RGB(*red, green, blue*). Pada konvolusi *layer, input* matriks pertama diperoleh berdasarkan tingkat warna yang ada pada masing-masing piksel sedangkan pada matriks kedua disesuaikan oleh *filter* yang digunakan peneliti. Pada saat training model, *input* gambar akan beresolusi 300x300x3, ini menunjukkan bahwa tinggi dan lebar piksel dari gambar sebesar 300 serta memiliki 3 *channel* yaitu RGB. *Filter* merupakan sebuah matriks lain yang memiliki tinggi dan lebar sama dan digunakan untuk menentukan pola objek deteksi lalu dikonvolusi atau dikalikan dengan nilai matriks *input*.

Setelah proses konvolusi selesai maka akan ada tahap yang bernama aktivasi ReLu (*Rectified Linear Unit*) yang berfungsi untuk menghilangkan nilai *negative* pada hasil keluaran. Semua yang ada pada hasil konvolusi bernilai *negative* akan diubah menjadi nol hingga *infinity*. Metode yang digunakan peneliti dalam proses *pooling layer* adalah *max-pooling.* Metode ini menghasilkan nilai maksimum saat sebuah filter dengan ukuran dan *stride* tertentu bergeser keseluruh area *feature map. Fully connected layer* merupakan proses transformasi pada dimensi data agar data tersebut dapat diklasifikasikan secara linear. Keluaran dari *fully conennected layer* yaitu berupa klasifikasi objek deteksi antara api dan asap. Langkah-langkah dalam melatih model CNN menggunakan *framework* TensorFlow akan dijelaskan sebagai berikut:

3.3.1 Konversi *File* XML ke CSV dan *Generate* TFRecord

Setelah klasifikasi data selesai maka akan menyimpan berkas yang berekstensi XML, sehingga perlu dikonversikan kedalam berkas yang berekstensi CSV agar dapat di-*generate TFRecord*. Proses *generate* TensorFlow *Record* digunakan untuk keperluan *feeding* data pada proses pelatihan.

3.3.2 Label Map

Tahap pelabelan adalah langkah untuk memberi keterangan pada saat hasil keluaran objek terdeteksi dan menyesuaikan jumlah kelas yang di-*input* kedalam *database*. Jumlah kelas pada *file* Label Map akan sama dengan *database* dan *generate TFRecord*.

3.3.3 Konfigurasi Pipeline

Langkah selanjutnya adalah konfigurasi *pipeline* yang berguna untuk mengatur algoritma dari *Convolutional Neural Network* (CNN) serta mengatur direktori *file training* dan pelabelan map. Pada kofigurasi *pipeline* ini pula dapat mengatur jumlah berapa banyak data yang akan di-*training* maupun yang akan dievaluasi dengan menggunakan *protobuf.*

3.3.4 *Training* Model

Langkah awal dalam proses *training* yaitu *Feeding* data pelatihan atau memasukkan data pelatihan ke dalam framework TensorFlow, lalu proses pelatihan dari data gambar untuk menghasilkan sistem pendeteksi asap dan api dengan menggunakan algoritma *Convolutional Neural Network*. Pada saat proses *training* model maka akan menghasilkan *checkpoint* yang dibuat secara otomatis oleh *framework* TensorFlow berbentuk *graph* tensor yang berguna untuk menyimpan informasi pada saat proses *training* model. Dalam proses komputasi menggunakan TensorFlow bisa menjadi sangat kompleks dan membingungkan, maka diperlukan Tensorboard yang berguna untuk memudahkan peneliti dan mengoptimalkan program TensorFlow serta memvisualisasikan *graph* TensorFlow.

3.3.5 *Export Graph*

Tujuan utama dalam proses *training* adalah mendapatkan sebuah model data yang dapat mendeteksi objek-objek yang diinginkan, maka setelah proses *training* selesai hasil dari *checkpoint* terakhir di *export* sehingga menjadi model data yang siap diujikan.

3.4 Evaluasi

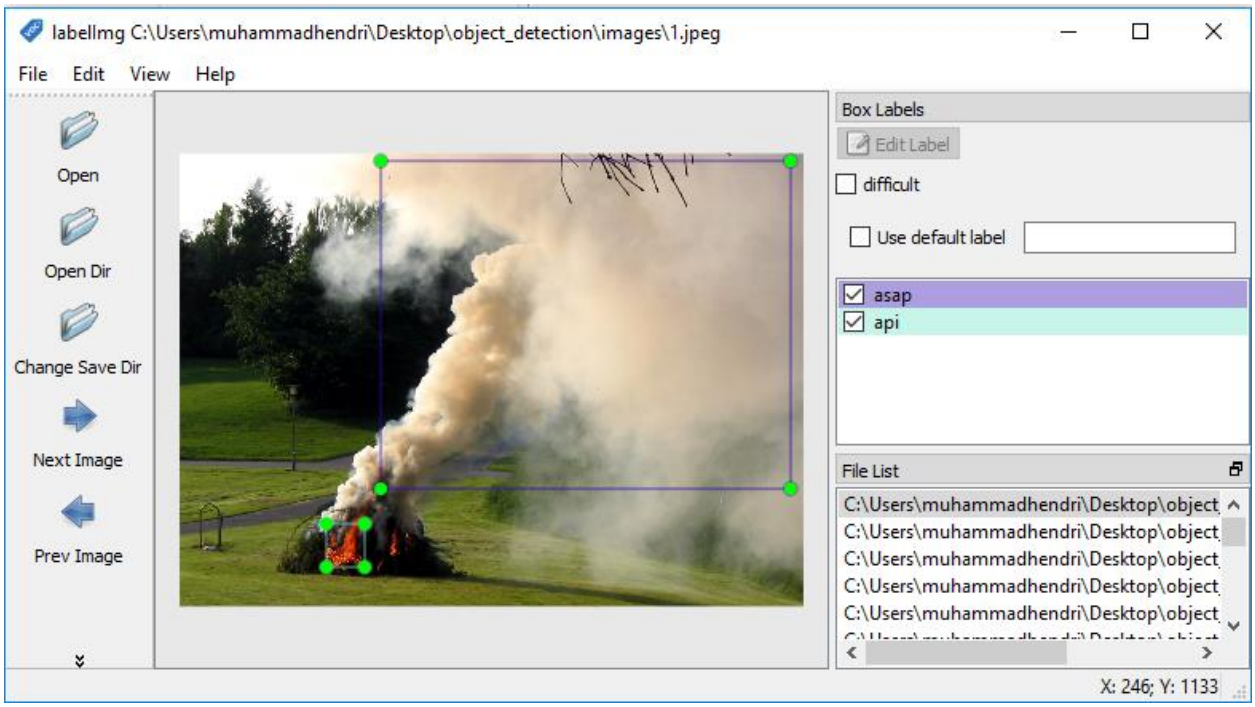
langkah terakhir adalah pengujian dari data yang telah di *export graph* dengan data uji. Jika tingkat akurasi dari pengujian gambar tinggi maka dilakukan interpretasi hasil dan pembahasan, dan jika tingkat akurasi dari pengujian gambar rendah maka dilakukan *training* model kembali. Dimana tingkat akurasi yang diinginkan peneliti sebagai objek deteksi asap dan api diatas dari 50%.

BAB 4
HASIL DAN PEMBAHASAN

Pada penelitian ini, peneliti melakukan klasifikasi menjadi dua kelas gambar yaitu api dan asap dengan menggunakan algoritma *Convolutional Neural Network* (CNN). Proses utama dalam pembuatan model ini diawali dengan proses *training* data. Proses ini bertujuan untuk membuat model dengan tingkat akurasi tinggi pada saat mendeteksi objek yang diinginkan. Parameter pengujian untuk mengukur tingkat keberhasilan model pendeteksi asap dan api adalah dengan melihat nilai akurasi dari objek gambar yang terdeteksi diatas dari 50% serta membandingkan jumlah data, iterasi dan *dropout regularization* saat proses *training* model. TensorFlow merupakan penelitian dari *Google* untuk mempermudah peneliti dalam *deeplearning, machine learning,* maupun *artificial intelligence.* Rancangan sistem yang digunakan peneliti adalah masukan(*input*), pengolahan citra dan keluaran(*output*).

4.1 Pelabelan Dataset Api dan Asap

Pelabelan gambar menggunakan program *labelImg.py* yang berguna untuk penentuan letak objek yang ingin dideteksi sebagai asap maupun api. Pelabelan ini dilakukan sebanyak 144 gambar yang berasal dari dataset peneliti. Berikut tampilan dari program *labelImg.py* :



Gambar 4.1 Pelabelan Dataset

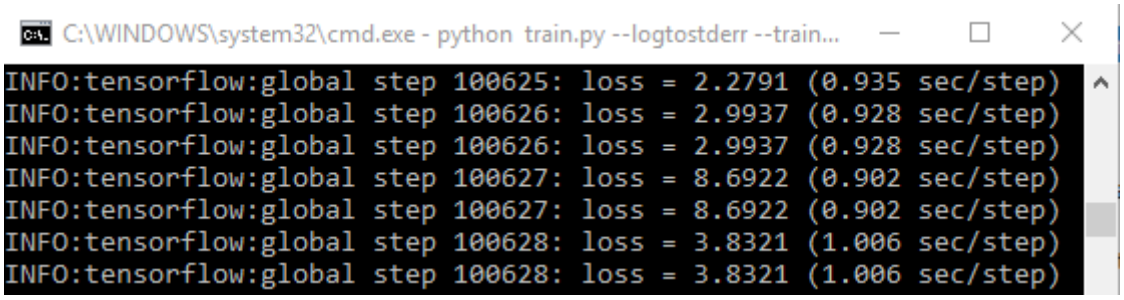
Pada gambar 4.1 menunjukkan bahwa kotak berwarna ungu adalah asap dan kotak bewarna biru adalah api. Semua gambar yang telah melalui proses pelabelan akan tersimpan dalam format XML.

4.2 Model Hasil *Training*

Pembahasan model *training* merupakan hasil implementasi dan proses pelatihan. Berikut adalah penjabaran mengenai proses model hasil *training* yang dilakukan oleh peneliti :

4.2.1 *Training Steps*

Training steps merupakan proses pelatihan yang berguna untuk melatih sistem agar mampu mengerti suatu pola(objek) yang diinginkan. Berdasarkan hasil penelitian, jumlah iterasi paling besar dilakukan hingga 101.557 langkah yang tersimpan pada *checkpoint folder*. Saat proses *training* model memerlukan waktu rata-rata 1.085-1.15 detik perdata untuk menggeneralisasi citra asap dan api. Namun di beberapa iterasi seperti pada iterasi ke 20.000 atau 60.000 mengalami peningkatan iterasi lebih cepat dibandingkan yang lainnya.

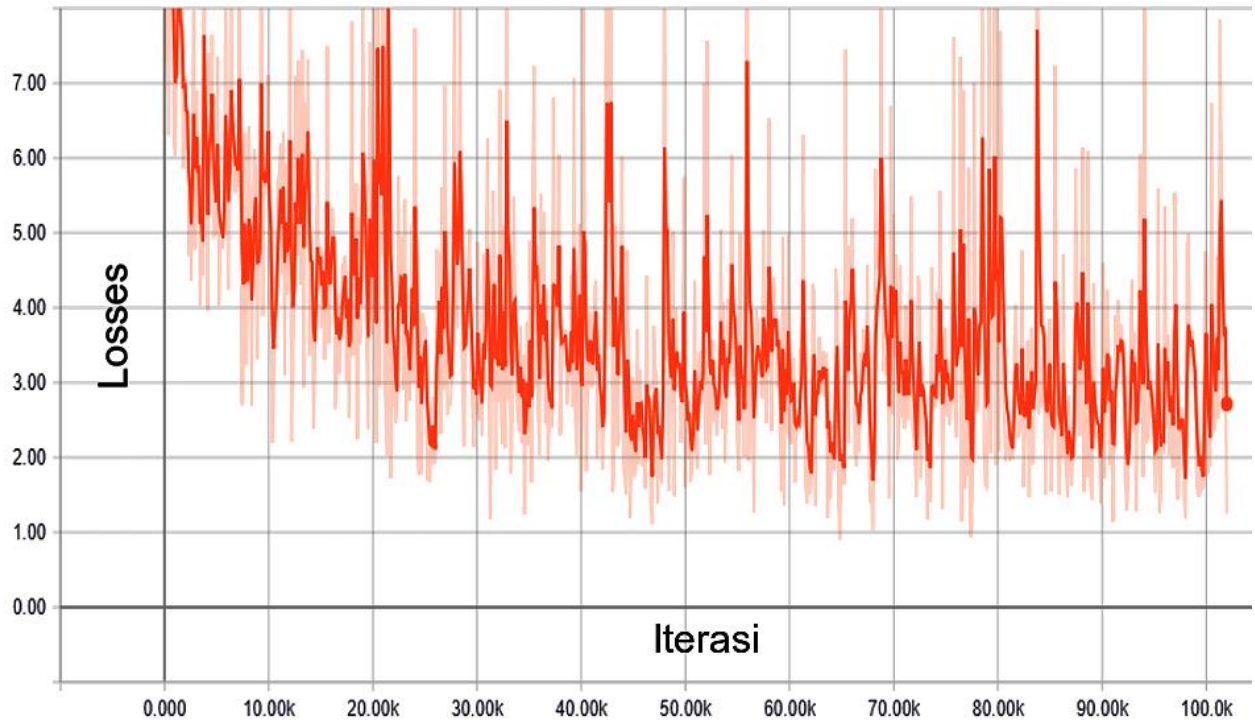


Gambar 4.2 *Training Step* Pada *Command Window*

Pada gambar 4.2 merupakan proses *training* yang terdapat pada *command window*. Dapat diketahui pada gambar 4.2 bahwa terdapat dua kelas yang sedang di *training* dan waktu yang diperlukan untuk proses pelatihan sekitar 0.8-1.2 detik perdata. Lamanya waktu pelatihan dapat dipengaruhi dari nilai *batch size*, peneliti menggunakan *batch size* pada proses *training* bernilai 2. Semakin besar resolusi gambar masukkan maka nilai *batch size* juga semakin besar dan memerlukan *memory* PC yang tinggi.

4.2.2 *Total Loss*

Pada saat proses *training* berjalan, semua proses data akan terekam dalam sebuah *file*. *File* tersebut dapat dilihat dengan menggunakan TensorBoard yang berbentuk seperti *local host*. Semua imformasi seperti *learning rate, num negative, num postives, target assignment, maupun total loss* tersimpan didalam *file* tersebut. Berikut hasil dari *total loss* pada proses *training* model :

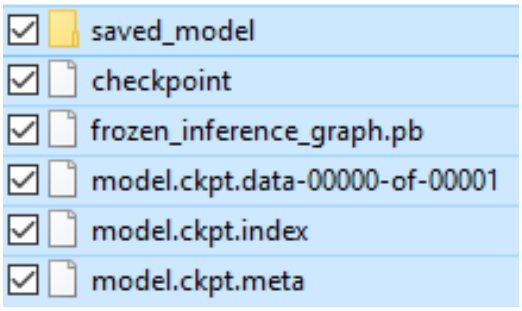


Gambar 4.3 Grafik *Total Loss*

Gambar 4.3 merupakan grafik *Total Loss* yang dihasilkan saat proses *training* berjalan hingga selesai. Pada umumnya nilai dari *Total Loss* dengan kriteria baik yaitu saat *steadystate* dibawah nilai 4. Pada gambar 4.3 diketahui bahwa hasil yang didapatkan berosilasi, yaitu dari 1.75-7.2. Hasil terakhir pada proses *training* pada langkah 101.557 mendapatkan nilai *losses* sebesar 2.715.

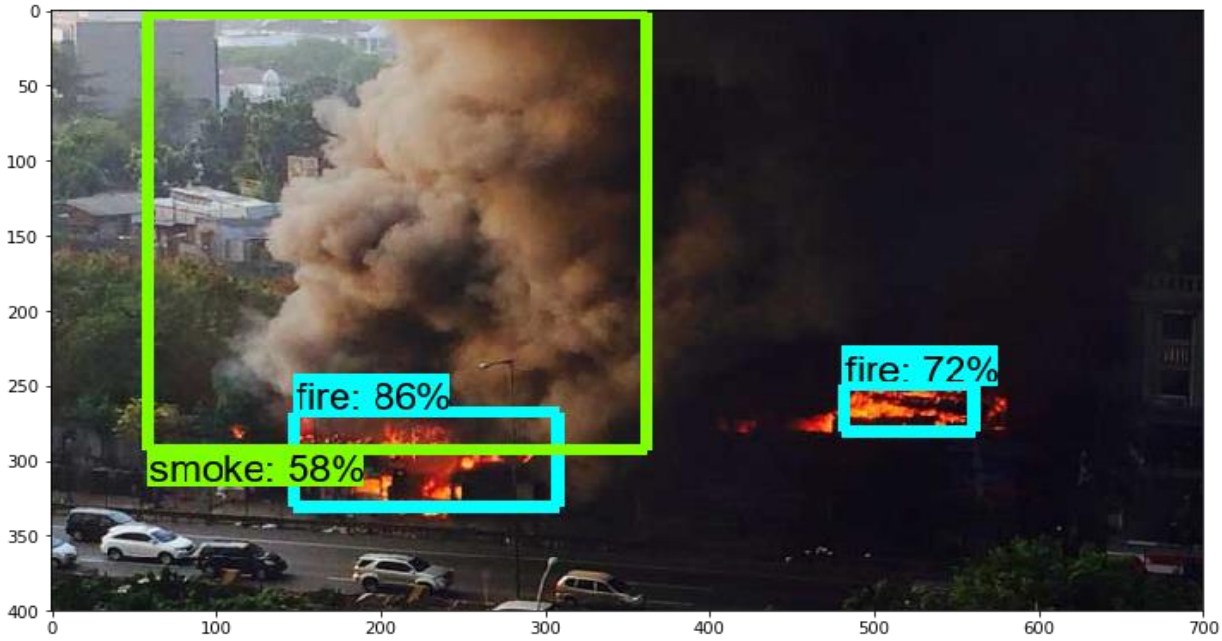
4.2.3 Model

Setelah proses *training* selesai maka akan menghasilkan *checkpoint folder*, selanjutnya meng-*export folder* tersebut untuk mendapatkan sebuah model data. Berikut isi dalam *folder* model tersebut :



Gambar 4.4 Hasil Model *Training*

4.3 Hasil Deteksi Asap dan Api



Gambar 4.5 Gambar Hasil *Training* Model

Pengujian model hasil *training* menggunakan 20 gambar dimana terdapat 10 gambar campuran (asap dan api), 1 gambar api, dan 9 gambar asap saja. Data pengujian diatur dengan format PNG dan beresolusi berbeda-beda. Hasil deteksi asap dan api dapat dilihat pada gambar 4.5 dimana objek api terdeteksi 86% dan 72%, sedangkan pada objek asap terdeteksi 58% serta memiliki resolusi sebesar 700x400 piksel. Rumus perhitungan objek yang terdeteksi asap dan api sebagai berikut :

$$N_{asap} = \frac{P}{19} x\ 100\% \quad (4.1)$$

$$N_{api} = \frac{I}{11} x\ 100\% \quad (4.2)$$

Keterangan :

N_{asap} = Persentase keberhasilan objek terdeteksi asap P = Jumlah objek terdeteksi asap

N_{api} = Persentase keberhasilan objek terdeteksi api I = Jumlah objek terdeteksi api

4.3.1 Hasil Deteksi Berdasarkan Jumlah Data Pelatihan

Pada pengujian ini dilakukan dengan perbandingan dari jumlah data yang akan di *training* yaitu 112, 128, dan 144 data dan diuji dengan 20 gambar dari asap dan api. Dimana seluruh pengujiannya dilakukan hingga 20.000 iterasi.

4.3.1.1 Pengujian Dengan 112 Data Dan 20.000 Iterasi

Tabel 4.1 Pengujian dengan 112 Data dan 20.000 Iterasi

Data ke-	Akurasi Asap (%)	Akurasi Api (%)	Deteksi Asap (19)	Deteksi Api (11)
1	50	-	✓	-
2	80	-	✓	-
3	×	×	×	×
4	72	×	✓	×
5	×	98	×	✓
6	-	93	-	✓
7	85	×	✓	×
8	50	-	✓	-
9	81	54	✓	✓
10	64	-	✓	-
11	99	-	✓	-
12	×	-	×	-
13	90	-	✓	-
14	63	×	✓	×
15	98	×	✓	×
16	50	50	✓	✓
17	×	×	×	×
18	×	-	×	-
19	56	-	✓	-
20	56	90	✓	✓

Ket :

✓ : Terdeteksi objek - : Tidak terdapat objek

× : *Error* pada deteksi objek

Pada Tabel 4.1 dapat diketahui bahwa objek yang terdeteksi sebagai asap ada 14 citra dengan persentase 73% dan objek terdeteksi sebagai api ada 5 citra dengan persentase 45%. Serta nilai rata-rata akurasi kemiripan asap 52% dan akurasi kemiripan api 35%.

4.3.1.2 Pengujian Dengan 128 Data Dan 20.000 Iterasi

Tabel 4.2 Pengujian dengan 128 Data dan 20.000 Iterasi

Data ke-	Akurasi Asap (%)	Akurasi Api (%)	Deteksi Asap (19)	Deteksi Api (11)
1	×	-	×	-
2	91	-	✓	-
3	50	×	✓	×
4	×	×	×	×
5	×	99	×	✓
6	-	92	-	✓
7	×	×	×	×
8	×	-	×	-
9	×	52	×	✓
10	×	-	×	-
11	×	-	×	-
12	×	-	×	-
13	×	-	×	-
14	×	64	×	✓
15	×	×	×	×
16	50	×	✓	×
17	69	×	✓	×
18	×	-	×	-
19	81	-	✓	-
20	×	96	×	✓

Ket :

✓ : Terdeteksi objek - : Tidak terdapat objek

× : *Error* pada deteksi objek

Pada Tabel 4.2 dapat diketahui objek yang terdeteksi sebagai asap ada 5 citra dengan persentase 26% sedangkan objek yang terdeteksi sebagai api ada 5 citra dengan persentase 45%. Serta nilai rata-rata akurasi kemiripan asap 17% dan akurasi kemiripan api 36%.

4.3.1.3 Pengujian Dengan 144 Data Dan 20.000 Iterasi

Tabel 4.3 Pengujian dengan 144 Data dan 20.000 Iterasi

Data ke-	Akurasi Asap (%)	Akurasi Api (%)	Deteksi Asap (19)	Deteksi Api (11)
1	80	-	✓	-
2	91	-	✓	-
3	79	×	✓	×
4	80	71	✓	✓
5	80	97	✓	✓
6	-	95	-	✓
7	85	×	✓	×
8	89	-	✓	-
9	83	69	✓	✓
10	87	-	✓	-
11	79	-	✓	-
12	88	-	✓	-
13	79	-	✓	-
14	82	×	✓	×
15	85	×	✓	×
16	50	50	✓	✓
17	79	×	✓	×
18	50	-	✓	-
19	75	-	✓	-
20	50	91	✓	✓

Ket :

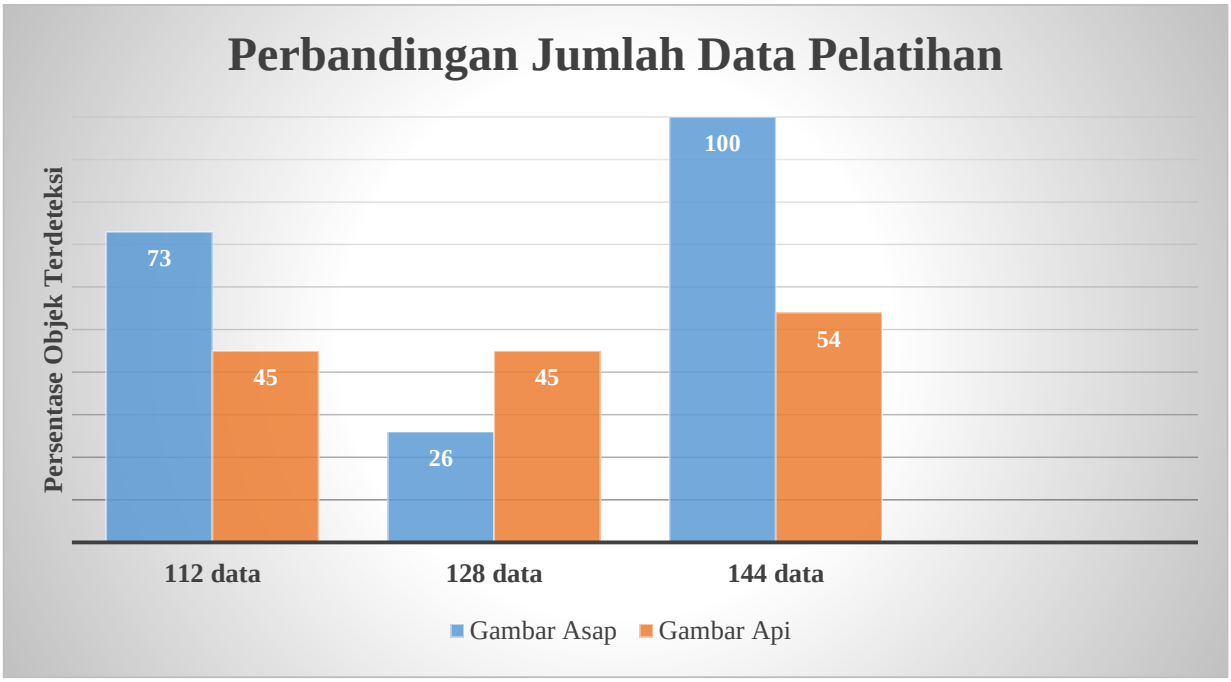
✓ : Terdeteksi objek - : Tidak terdapat objek

× : *Error* pada deteksi objek

Berdasarkan Tabel 4.3 bahwa hasil yang terdeteksi sebagai asap ada 19 citra sehingga persentase objek terdeteksi asap 100%, sedangkan hasil yang terdeteksi sebagai api ada 6 citra sehingga persentasenya 54%. Dengan nilai rata-rata akurasi kemiripan asap 77% dan akurasi kemiripan api 43%.

4.3.1.4 Grafik Perbandingan Jumlah Data Pelatihan

Berikut grafik dari perbandingan jumlah data tersebut:



Gambar 4.6 Perbandingan Jumlah Data Pelatihan

Berdasarkan gambar 4.6 didapatkan hasil yang cukup baik pada 144 data. Dimana pada model tersebut memiliki persentase objek terdeteksi asap sebesar 100% dan pada citra api terdeteksi 54%. Pada model yang menggunakan 128 data mendapatkan hasil terburuk dengan *error* yang cukup besar sehingga banyak gambar pengujian yang tidak terdeteksi sebagai asap dan api. akurasi kemiripan asap dan api dalam perbandingan jumlah data pelatihan memiliki hasil yang cukup tinggi pada model pelatihan menggunakan 144 data.

4.3.2 Hasil Deteksi Berdasarkan Jumlah Iterasi Pelatihan

Pada pengujian ini dilakukan dengan perbandingan dari jumlah iterasi yang akan di *training* yaitu 20.000, 60.000, dan 100.000 langkah dan diuji dengan 20 gambar dari asap dan api. Dimana seluruh pengujiannya menggunakan 144 data.

4.3.2.1 Pengujian Dengan 60.000 Iterasi

Tabel 4.4 Pengujian dengan 60.000 Iterasi dan 144 Data

Data ke-	Akurasi Asap (%)	Akurasi Api (%)	Deteksi Asap (19)	Deteksi Api (11)
1	51	-	✓	-
2	99	-	✓	-
3	53	×	✓	×
4	66	×	✓	×
5	×	57	×	✓
6	-	66	-	✓
7	83	×	✓	×
8	75	-	✓	-
9	91	92	✓	✓
10	62	-	✓	-
11	×	-	×	-
12	86	-	✓	-
13	74	-	✓	-
14	55	×	✓	×
15	75	×	✓	×
16	×	50	×	✓
17	78	92	✓	×
18	94	-	✓	-
19	99	-	✓	-
20	58	97	✓	✓

Ket :

✓ : Terdeteksi objek - : Tidak terdapat objek

× : *Error* pada deteksi objek

Pada Tabel 4.4 dapat diketahui objek yang terdeteksi sebagai asap ada 16 citra dengan persentase 84% sedangkan objek yang terdeteksi sebagai api ada 5 citra dengan persentase 45%. Serta nilai rata-rata akurasi kemiripan dari asap 63% dan akurasi kemiripan api 41%.

4.3.2.2 Pengujian Dengan 100.000 Iterasi

Tabel 4.5 Pengujian dengan 100.000 Iterasi dan 144 Data

Data ke-	Akurasi Asap (%)	Akurasi Api (%)	Deteksi Asap (19)	Deteksi Api (11)
1	99	-	✓	-
2	100	-	✓	-
3	99	50	✓	✓
4	98	95	✓	✓
5	×	58	×	✓
6	-	90	-	✓
7	99	×	✓	×
8	99	-	✓	-
9	90	89	✓	✓
10	99	-	✓	-
11	99	-	✓	-
12	99	-	✓	-
13	99	-	✓	-
14	99	×	✓	×
15	75	×	✓	×
16	50	50	✓	✓
17	99	89	✓	✓
18	100	-	✓	-
19	99	-	✓	-
20	90	83	✓	✓

Ket :

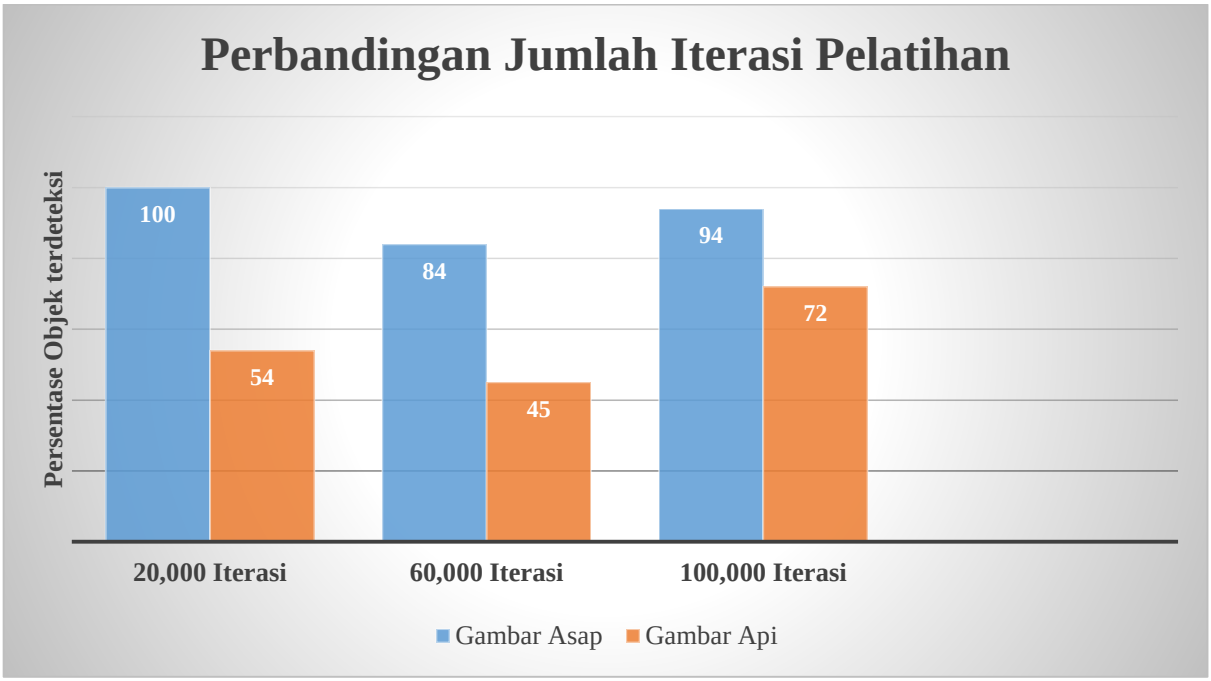
✓ : Terdeteksi objek - : Tidak terdapat objek

× : *Error* pada deteksi objek

Pada tabel 4.5 diketahui bahwa hasil deteksi objek sebagai asap ada 18 citra dengan persentase 94% dan deteksi objek sebagai api ada 8 citra dengan persentase 72%. Serta nilai rata-rata akurasi kemiripan dari asap 89% dan akurasi kemiripan api 54%.

4.3.2.3 Perbandingan Jumlah Iterasi Pelatihan

Berikut grafik perbandingan jumlah iterasi:



Gambar 4.7 Perbandingan Jumlah Iterasi Pelatihan

Pada gambar 4.7 dapat diketahui bahwa hasil deteksi pada asap mendapatkan nilai terbaik pada iterasi 20.000 yaitu 100% sedangkan hasil deteksi api mendapatkan nilai terbaik pada iterasi ke 100.000 dengan nilai 72%. Model CNN yang telah dibuat cukup baik dalam mengklasifikasikan citra asap dan api. Semakin besar jumlah iterasi yang digunakan maka nilai akurasi kemiripan dari objek deteksi akan semakin besar. Hal ini menunjukan bahwa sebuah *machine learning* lebih banyak memahami pola citra sehingga, ketepatan dalam proses klasifikasi akan semakin baik.

4.3.3 Hasil Deteksi Menggunakan *Dropout Regularization*

Pengujian ini menggunakan 20 gambar seperti pengujian sebelumnya dan menggunakan 144 data *training* serta iterasi hingga 20.000 langkah. Yang berbeda dari pengujian sebelumnya adalah pada program *pipeline* nilai *dropout* diberi masukan *true*.

4.3.3.1 Pengujian Dengan *Dropout Regularization*

Tabel 4.6 Pengujian menggunakan *Dropout* dengan 20.000 Iterasi dan 144 Data

Data ke-	Akurasi Asap (%)	Akurasi Api (%)	Deteksi Asap (19)	Deteksi Api (11)
1	95	-	✓	-
2	77	-	✓	-
3	58	86	✓	✓
4	81	90	✓	✓
5	×	96	×	✓
6	-	96	-	✓
7	62	55	✓	✓
8	75	-	✓	-
9	87	98	✓	✓
10	74	-	✓	-
11	88	-	✓	-
12	90	-	✓	-
13	90	-	✓	-
14	59	71	✓	✓
15	90	57	✓	✓
16	50	50	✓	✓
17	56	68	✓	✓
18	96	-	✓	-
19	86	-	✓	-
20	58	95	✓	✓

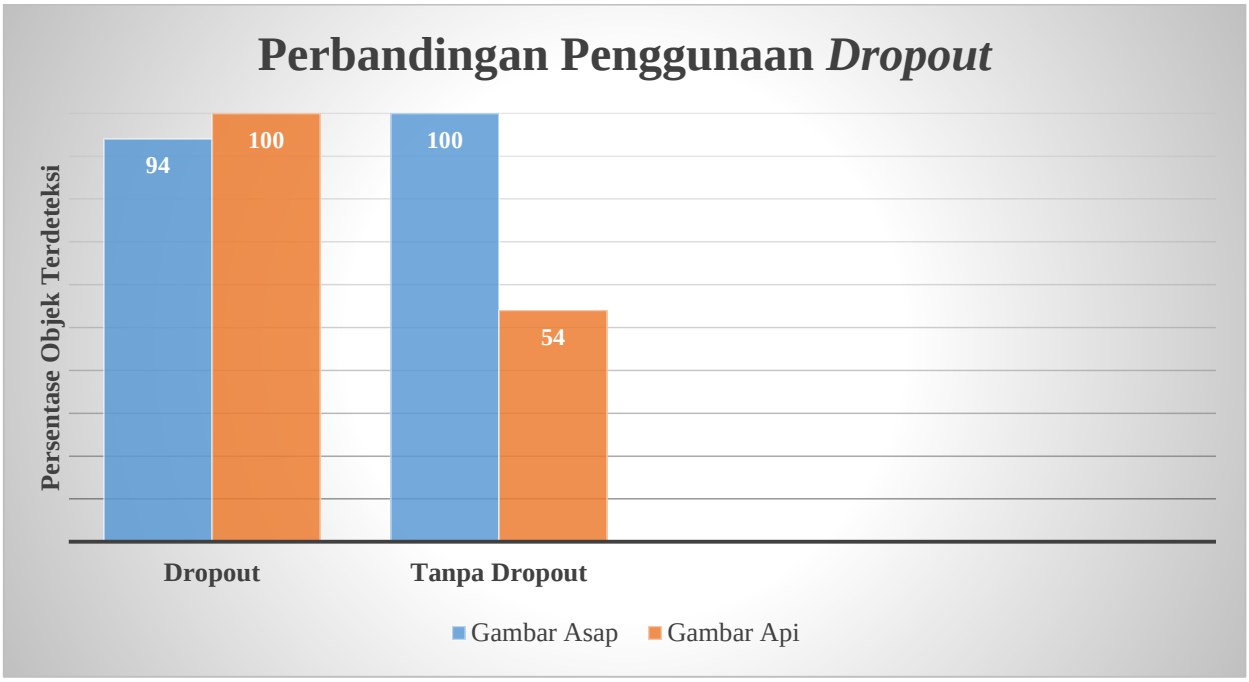
Ket :

✓ : Terdeteksi objek - : Tidak terdapat objek

× : Error pada deteksi objek

Pada tabel 4.6 dapat diartikan bahwa pengujian model menggunakan *dropout regularization* mendapatkan hasil yang baik yaitu dengan nilai deteksi asap sebesar 94% dimana objek yang terdeteksi sebagai asap ada 18 citra dan deteksi api sebesar 100% dimana objek terdeteksi sebagai api ada 11 citra. Serta nilai rata-rata akurasi kemiripan dari asap sebesar 72% dan rata-rata akurasi kemiripan api sebesar 78%.

4.3.3.2 Perbandingan Tanpa Dan Menggunakan *Dropout Regularization*



Gambar 4.8 Perbandingan Penggunaan *Dropout Regularization*

Dapat dilihat pada gambar 4.8 bahwa hasil yang didapat menggunakan *dropout* memiliki hasil yang cukup memuaskan, yaitu dengan nilai persentase asap sebesar 94% dan nilai persentase api sebesar 100%. Sedangkan tanpa adanya *dropout* persentase dari api menurun yaitu sebesar 54%. Mengaktifkan nilai *dropout regularization* maka akan membuat algoritma *Convolutional Neural Network* menjadi lebih teratur, oleh sebab itu hasil yang didapat menjadi lebih akurat.

BAB 5
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil penelitian yang dilakukan dapat disimpulkan sebagai berikut:

1. Perancangan sistem deteksi asap dan api didapatkan model sistem berdasarkan jumlah data yaitu pada 144 data *input.* Dengan hasil 100% objek deteksi pada asap dan 54% objek deteksi pada api. Jika berdasarkan jumlah iterasi, maka *output* akan lebih akurat mendeteksi objek bila jumlah iterasi semakin besar.
2. Jika *training* model menggunakan *dropout regularization* maka algoritma CNN akan lebih teratur dalam mempelajari model sistem, sehingga akan lebih akurat dalam mendeteksi objek.
3. Hasil terbaik penelitian ini yaitu saat menggunakan *dropout regularization* dengan persentase hasil akurasi asap sebesar 94% dan api sebesar 100%.
4. Dalam *training* model CNN nilai *batch size* sangat berpengaruh dalam proses komputasi dan waktu yang dibutuhkan saat mempelajari suatu model objek deteksi.

5.2 Saran

Saran yang dapat diberikan pada penelitian selanjutnya:

1. Untuk mendapatkan hasil yang lebih baik maka diperlukan komputer dengan GPU yang mumpuni dalam meningkatkan nilai *batch size.*
2. Mencoba memperbanyak iterasi dan jumlah data *input* dengan menggunakan *dropout regularization*.
3. Mencoba untuk mendeteksi objek asap dan api menggunakan *video input* maupun *real time video processing.*

DAFTAR PUSTAKA

[1] S. J. Chen, D. C. Hovde, K. A. Peterson, and A. W. Marshall, “Fire detection using smoke and gas sensors,” *Fire Saf. J.*, vol. 42, no. 8, pp. 507–515, 2007.

[2] G. Xu, Y. Zhang, Q. Zhang, G. Lin, and J. Wang, “Deep domain adaptation based video smoke detection using synthetic smoke images,” *Fire Saf. J.*, vol. 93, pp. 53–59, Oct. 2017.

[3] D. Ekorianto, “Perancangan Sistem Pengenalan Wajah Untuk Mendukung Smart Home System,” Universitas Islam Indonesia, 2017.

[4] D. k Appana, “A video-based smoke detection using smoke flow pattern and spatial-temporal energy analyses for alarm systems,” *Inf. Sci. (Ny).*, vol. 418–419, pp. 91–101, Dec. 2017.

[5] I. Taufiq, “Deep Learning Untuk Deteksi Tanda Nomor Kendaraan Bermotor Menggunakan Algoritma Convolutional Neural Network Dengan Python,” 2018.

[6] Murinto and S. Hartati, “Analisis Citra Untuk Pengenalan Fitur Pada Perangkat Sistem Informasi Geografis,” 2016.

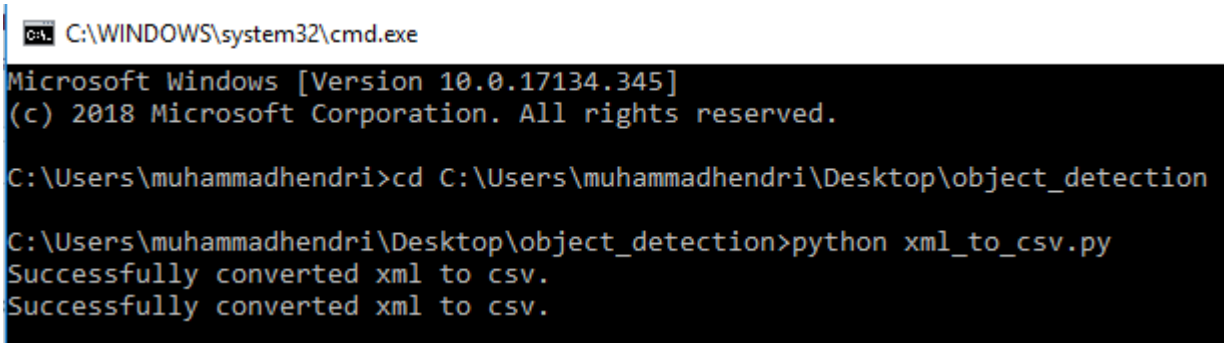
[7] B. B. Traore, B. Kamsu-Foguem, and F. Tangara, “Deep convolution neural network for image recognition,” *Ecol. Inform.*, p. #pagerange#, 2018.

[8] S. J. Lee, T. Chen, L. Yu, and C. H. Lai, “Image Classification Based on the Boost Convolutional Neural Network,” *IEEE Access*, vol. 6, no. c, pp. 12755–12768, 2018.

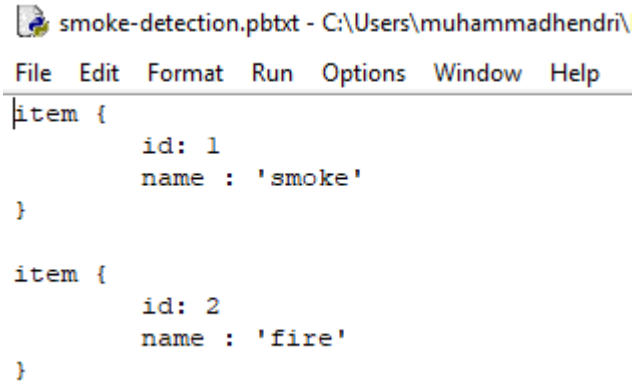
[9] G. Hinton, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” vol. 15, pp. 1–30, 2014.

LAMPIRAN

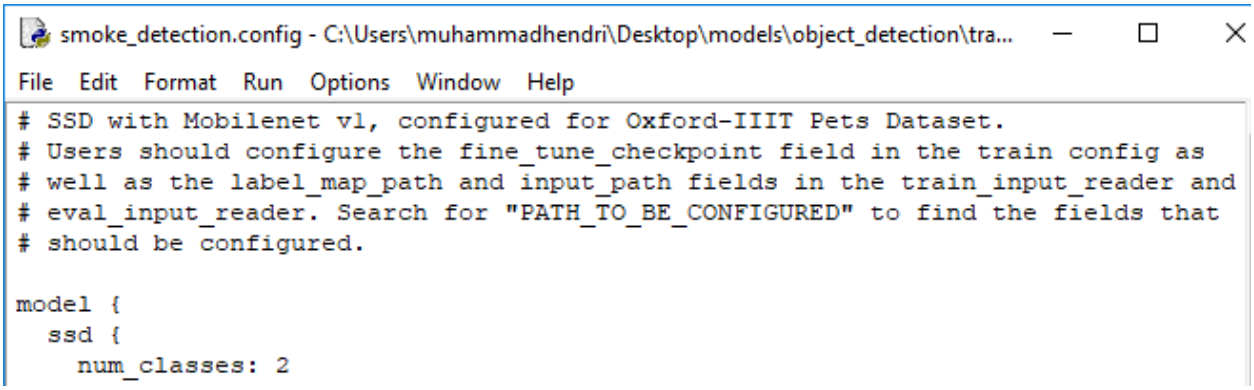
1. Konversi XML ke CSV



2. Label Map



3. Konfigurasi Pipeline



4. Hasil Deteksi Asap dan Api


fire: 9
smoke: 81%


fire: 96%

5. Learning Rate