PERANCANGAN SISTEM DETEKSI ASAP DAN API MENGGUNAKAN PEMROSESAN CITRA

Muhammad Hendri[1], Elvira Sukma Wahyuni[2]

Jurusan Teknik Elektro, Universitas Islam Indonesia
Jl Kaliurang KM 14.5 Yogyakarta, Indonesia
14524051@students.uii.ac.id
155231301@uii.ac.id

***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*.

I. PENDAHULUAN

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 *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.

II. TINJAUAN PUSTAKA

A. 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 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[3].

Penelitian mengenai *deep learning* untuk deteksi tanda nomor kendaraan bermotor dengan algortima *convolutional neural network* dengan *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[4].

B. 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 vetikal 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.

C. Pengolahan Citra

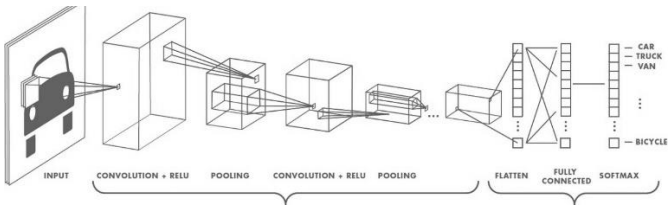
Pengolahan citra merupakan suatu proses pengolahan data menggunakan komputer sehingga citra tersebut dapat menjadi sebuah citra yang memiliki kualitas baik. Tujuan dari pengolahan citra ini adalah untuk memperbaiki kualitas suatu citra agar dapat diinterpretasi dengan manusia maupun komputer. Pengekstrasian 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[5].

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, Activation(*Relu), *pooling layer, fully connected layer* sehingga semua layer tersebut akan dikonvolusikan secara linear.

D. 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*[6].



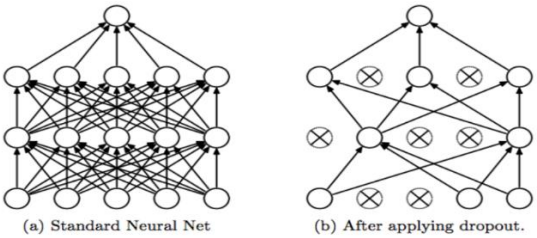
Gambar 2.1 CNN *layer*

E. 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*[7].

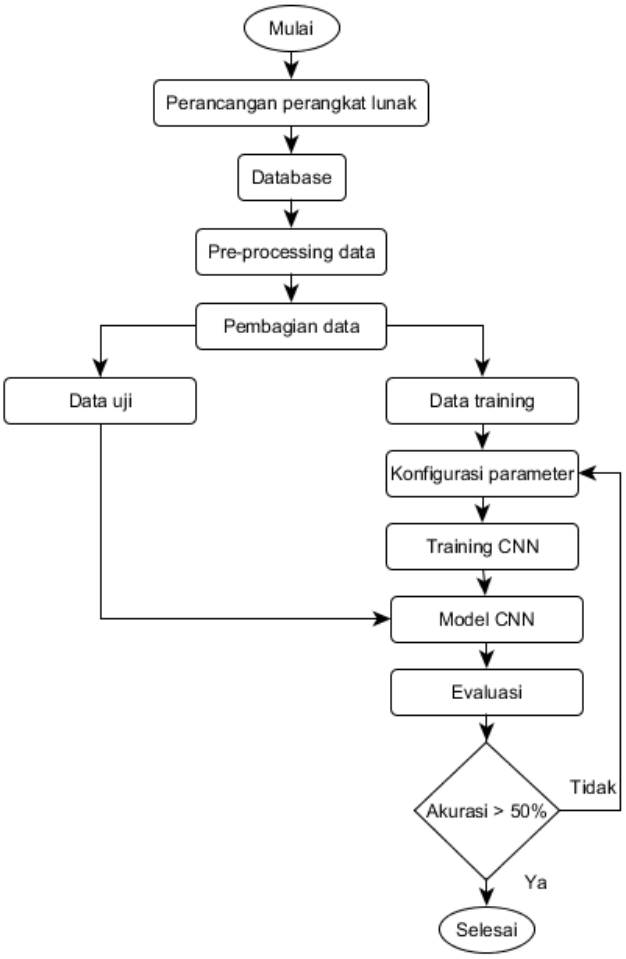


Gambar 2.2 *Dropout regularization*

III. METODE PENELITIAN

A. *Alur Penelitian*

Metode yang digunakan dalam mendeteksi asap dan api menggunakan algoritma *convolutional neural network* dapat dilihat pada bagan alir gambar 3.1.



Gambar 3.1 Metode Penelitian

B. *Perancangan Perangkat Penelitian*

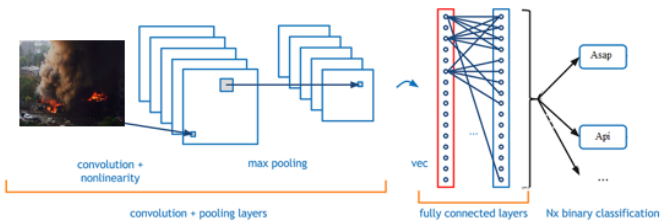
Sistem deteksi asap dan api ini memerlukan perangkat keras berupa *personal computer* (PC). Sistem ini mendeteksi objek secara gambar. *Personal computer* (PC) digunakan untuk memonitor serta sebagai pengendali sistem. Perangkat lunak yang digunakan dalam penelitian ini yaitu *Python-3, Visual Studio* 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.*

C. *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.

D. *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 Asap dan Api 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:

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.

2. Label Map

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

3. Konfigurasi Pipeline

Langkah selanjutnya adalah konfigurasi *pipeline* yang berguna untuk mengatur algoritma dari *Convolutional Neural Network* (CNN) serta mengatur direktori *file* latihan 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*.

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.

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.

E. Evaluasi

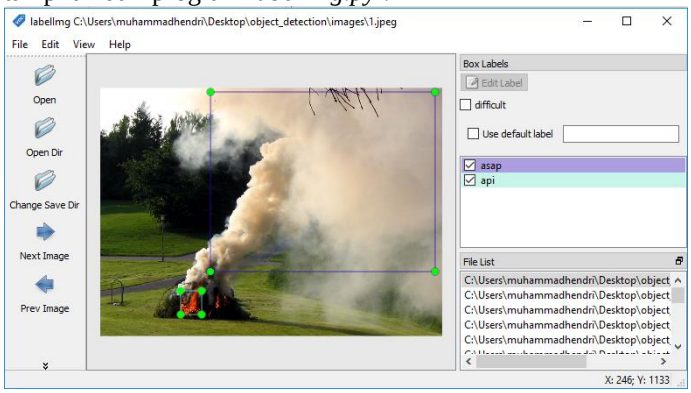
langkah terakhir adalah pengujian dari data yang telah di *export graph* dengan data uji. Jika tingkat akurasi dari pengujian gambar tinggi maka dilakukan interpretasi hasil dan pembahasan, dan jika tingkat akurasi dari pengujian gambar rendah maka dilakukan *training* model kembali. Dimana tingkat akurasi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

IV. HASIL DAN ANALISIS

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 mengaktifkan *dropout regularization* saat proses *training* model.

A. Pelabelan Dataset Asap dan Api

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.

B. Model Training

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

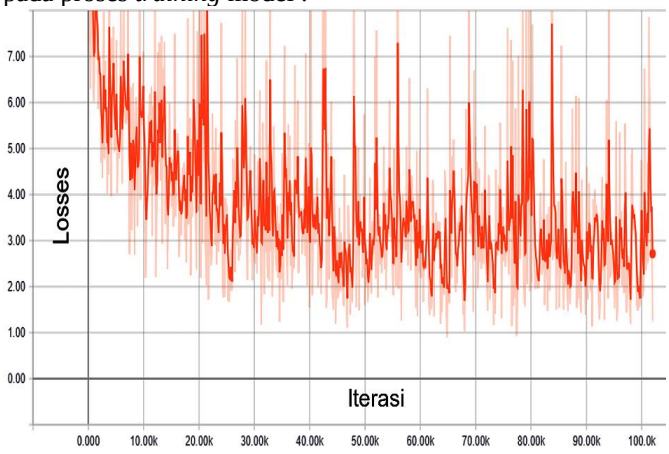
1. *Training Steps*

Training steps merupakan proses pelatihan yang berguna untuk melatih sistem agar mampu mengerti suatu

pola(objek) yang diinginkan. Berdasarkan hasil *training* yang dilakukan oleh peneliti sebanyak 101.557 langkah yang tersimpan pada *checkpoint* folder. Saat *training* model memerlukan waktu rata-rata 1.085-1.15 detik perdata. Namun di beberapa iterasi seperti pada iterasi ke 20.000 atau 60.000 mengalami kecepatan iterasi lebih cepat dibandingkan yang lainnya.

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 :

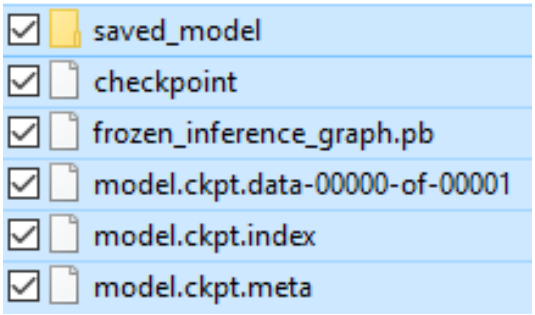


Gambar 4.2 Grafik *Total Loss*

Gambar 4.2 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 101557 mendapatkan nilai *losses* sebesar 2.715.

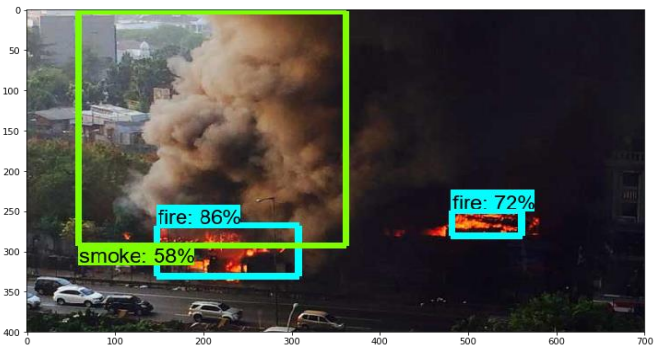
3. *Model*

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 4.3 Model Hasil *Training*

C. Hasil Deteksi Asap dan Api



Gambar 4.4 Hasil Deteksi Asap dan Api

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 4.4 dimana objek api terdeteksi 86% dan 72%, sedangkan pada objek asap terdeteksi 58% serta memiliki resolusi sebesar 700x400 piksel. Seluruh *database* yang digunakan dalam penelitian ini memerlukan citra asap dan api yang besar dan tebal agar pengolahan citra (*image processing*) yang dihasilkan memiliki kualitas yang baik. Pengujian ini menggunakan rumus perhitungan akurasi 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

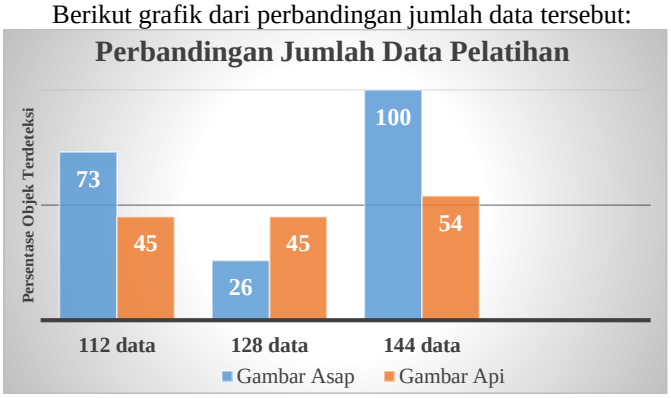
P = Jumlah objek terdeteksi asap

N_{api} = Persentase keberhasilan objek terdeteksi api

I = Jumlah objek terdeteksi api

1. *Grafik Perbandingan Jumlah Data Pelatihan*

Berikut grafik dari perbandingan jumlah data tersebut:



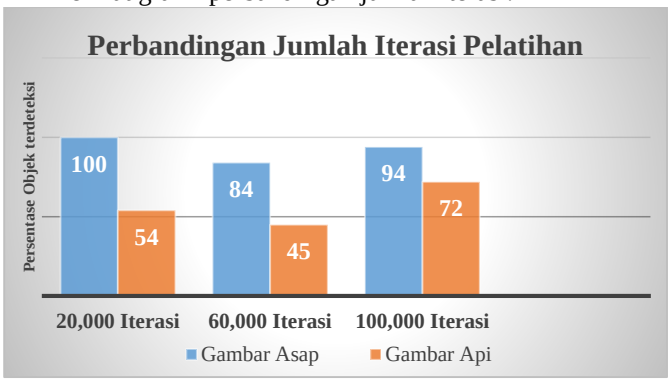
Gambar 4.5 Grafik Perbandingan Jumlah Data

Berdasarkan gambar 4.5 didapatkan hasil yang cukup baik pada 144 data. Dimana pada model tersebut mendapatkan nilai akurasi asap sebesar 100% dan pada gambar 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 memiliki hasil yang cukup tinggi pada model pelatihan 144 data.

2. *Berdasarkan Jumlah Iterasi Pelatihan*

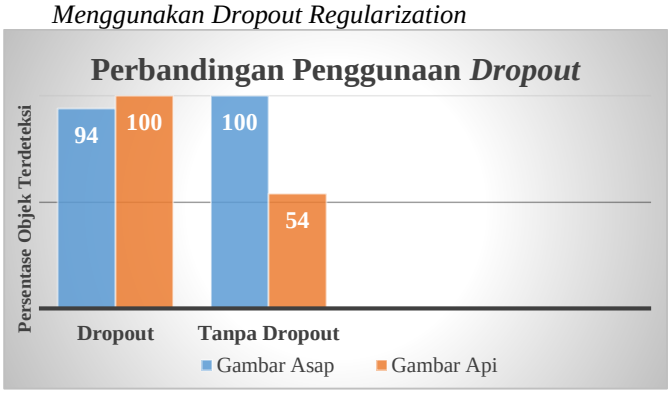
Berikut grafik perbandingan jumlah iterasi:



Gambar 4.6 Grafik Perbandingan Iterasi

Pada gambar 4.6 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.

3. *Perbandingan Tanpa Dropout Regularization Dan Menggunakan Dropout Regularization*



Gambar 4.7 Grafik perbandingan *Dropout Regularization*

Dapat dilihat pada gambar 4.7 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* persentasi dari api menurun yaitu sebesar 54%. Mengaktifkan nilai *dropout* maka akan membuat algoritma *Convolutional Neural Network* menjadi lebih teratur, oleh sebab itu hasil yang didapat menjadi lebih akurat.

V. KESIMPULAN

Berdasarkan hasil penelitian yang dilakukan dapat disimpulkan sebagai berikut:

1. Perancangan sistem deteksi asap dan api didapatkan model sistem terbaik 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.

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. 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.

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

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

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

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