

Sistem Deteksi Infeksi COVID-19 Pada Hasil X-Ray Rontgen menggunakan Algoritma Convolutional Neural Network (CNN)

Muhammad Saiful^{1*}, LM.Samsu², Fathurrahman³

^{1,2,3}Program Studi Sistem Informasi, Universitas Hamzanwadi

*Saipulslbm@gmail.com

Abstrak

Perkembangan teknologi dunia semakin berkembang pesat, khususnya pada bidang kesehatan yang berupa alat pendeteksian berbagai macam objek, termasuk objek penyakit. Teknologi yang di maksud adalah bagian dari kecerdasan buatan yang mampu mengenal sekumpulan citra dan mengklasifikasikan secara otomatis dengan teknik deep learning. Salah satu jaringan deep learning yang banyak digunakan adalah convolutional neural network dengan teknologi computer vision. Salah satu permasalahan pada computer vision yang masih berkembang adalah object detection sebagai teknologi yang berguna untuk mengenal objek pada gambar selayaknya manusia mengenal objek gambar. Dalam hal tersebut, sebuah mesin komputer di latih pembelajaran dengan menggunakan jaringan saraf tiruan. Salah satu sub tipe jaringan saraf tiruan yang mampu menangani permasalahan computer vision tersebut adalah dengan menggunakan teknik deep learning dengan algoritma convolutional neural network. Tujuan dari penelitian ini adalah mengetahui bagaimana perancangan sistem, arsitektur jaringan yang digunakan untuk pendeteksian infeksi COVID-19. Sistem ini tidak dapat melakukan pendeteksian objek yang lain. Hasil deteksi infeksi COVID-19 dengan algoritma convolutional neural network menunjukkan nilai akurasi yang unlimited yaitu berkisar di angka 60-99%...

Kata Kunci : *Computer Vision, Deep Learning, Convolutional Neural Network*

Abstract

The development of the world's technology is growing rapidly, especially in the field of health in the form of detection tools of various objects, including disease objects. The technology in point is part of artificial intelligence that is able to recognize a set of imagery and classify automatically with deep learning techniques. One of the deep learning networks widely used is convolutional neural network with computer vision technology. One of the problems with computer vision that is still developing is object detection as a useful technology to recognize objects in the image as if humans knew the object of the image. In this case, a computer machine is trained in learning using artificial neural networks. One of the sub types of artificial neural networks that are able to handle computer vision problems is by using deep learning techniques with convolutional neural network algorithms. The purpose of this research is to find out how to design the system, the network architecture used for COVID-19 infection detection. The system cannot perform detection of other objects. The results of COVID-19 infection detection with convolutional neural network algorithm show unlimited accuracy value that ranges from 60-99%.

Keywords: *Computer Vision, Deep Learning, Convolutional Neural Network*

1. Pendahuluan

COVID-19 (corona virus disease 2019) atau dikenal juga dengan virus corona adalah virus yang menyerang sistem pernapasan. Penyakit

karena infeksi virus ini disebut severe acute respiratory syndrome corona virus 2 (SARS-CoV-2). COVID-19 adalah virus jenis baru yang menular ke manusia. Walaupun lebih

bayak menyerang lansia. virus ini sebenarnya bisa menyerang siapa saja, mulai dari bayi, anak-anak, hingga orang dewasa, termasuk ibu hamil dan ibu menyusui. COVID-19 pertama kali ditemukan di kota Wuhan, China pada akhir Desember 2019. Virus ini menular dengan sangat cepat dan telah menyebar ke hampir semua negara, termasuk Indonesia, hanya dalam waktu beberapa bulan.

Terdapat beberapa metode dalam mendeteksi dan mengenali objek pada sebuah image menggunakan jaringan saraf tiruan, salah satunya adalah algoritma Convolutional Neural Network (CNN) yang sering digunakan pada pengolahan data image. Convolutional Neural Network (CNN) merupakan salah satu metode yang terdapat dalam deep learning yang banyak digunakan untuk menyelesaikan permasalahan yang berkaitan dengan object detection dan image classification. Object detection (pendeteksian objek) baru-baru ini menjadi salah satu bidang yang paling menarik dalam computer vision dan artificial intelligence (AI). Pendeteksian objek merupakan teknologi komputer yang berkaitan dengan computer vision dan image processing yang berhubungan dengan mendeteksi suatu objek dalam citra digital yang dapat berupa warna dan bentuk objek.

Convolutional Neural Network (CNN) banyak digunakan pada penelitian terdahulu karena

memiliki tingkat akurasi yang relative tinggi dan memiliki hasil yang signifikan dalam pengenalan citra. Salah satu software yang dapat dimanfaatkan untuk kepentingan ini adalah bahasa pemrograman Python dan library TensorFlow, yang termasuk kedalam software artificial intelligence untuk mendeteksi berbagai macam objek berdasarkan dataset gambar (image).

Berdasarkan uraian di atas, dalam penelitian ini dibuat sebuah sistem untuk mendeteksi antara hasil x-ray rontgen foto thorax terinfeksi COVID-19 dan normal pada sekumpulan gambar. Adapun algoritma yang digunakan pada sistem adalah algoritma Convolutional Neural Network. Oleh karena itu, peneliti membuat penelitian yang berjudul "Sistem Deteksi Infeksi COVID-19 pada Hasil X-Ray Rontgen menggunakan Algoritma Convolutional Neural Network (CNN)". Dengan adanya sistem diharapkan dapat membantu tenaga medis ataupun pemerintah yang menangani masalah COVID-19.

2. Tinjauan Pustaka

2.1. Penelitian Terkait

Dalam penulisan skripsi ini, penulis terinspirasi menggunakan referensi dari penelitian-penelitian sebelumnya yang berkaitan dengan pembahasan skripsi yang dibuat. Penelitian ini terkait penggunaan data mining dengan menggunakan algoritma K-Nearest Neighbor. Berikut ini penelitian terdahulu yang berkaitan dengan proposal skripsi antara lain:

Penelitian yang dilakukan (Moh. Farid Wajdi and Jagat Sugiantara, 2018) dengan judul "Pemanfaatan Teknik Pengenalan Wajah Berbasis Opencv untuk Sistem Informasi Pencatatan Kehadiran Dosen". Pada sistem yang diteliti oleh dosen Universitas Hamzanwadi Lombok Timur ini dikemukakan bahwa sistem pencatatan kehadiran saat diujikan membuktikan Per-sepsi Responden terhadap variabel-variabel pendukung metode TAM menunjukkan hasil interpretasi bernilai BAIK yang dibuktikan dengan rata-rata total skor tiap indikator variabel penyusunnya yaitu rata-rata total skor diatas 4.00. dimana interpretasi BAIK memiliki rentang nilai 3.68 – 5.00 [1].

Penelitian yang dilakukan oleh (Imam Fathurrahman and Indra Gunawan, 2018) dengan judul "Pengenalan Citra Logo Kendaraan Menggunakan Metode Gray Level Co-Occurrence Matrix (GlcM) dan Jst-Backpropagation". Pada sistem yang diteliti oleh dosen Universitas Hamzanwadi Lombok Timur ini dikemukakan bahwa Penentuan parameter dalam jaringan syaraf tiruan backpropagation akan memberikan hasil klasifikasi yang optimal adalah; 14 input layer, 7 node hidden layer, 4 node output, dan learning rate 0.05 menghasilkan eror target 0.018391 pada epoch ke 43 dari 49 epoch sehingga tingkat keberhasilan pengenalan logo dengan arah 00 ekstraksi fitur GLCM menggunakan JST- Backpropagation adalah 95.7% dalam waktu 20 detik [2].

Penelitian yang dilakukan oleh (Imam Fathurrahman, Amri Muliawan Nur, and Fathurrahman, 2018) yang berjudul "Identifikasi Kematangan Buah Mentimun Berbasis Citra Digital Menggunakan Jaringan Syaraf Tiruan Backpropagation". Pada sistem yang diteliti oleh dosen Universitas Hamzanwadi Lombok Timur ini dikemukakan bahwa penelitian ini mampu mencapai akurasi yang baik sebesar 89.6%. Pada penelitian-penelitian selanjutnya, tingkat akurasi ini masih bisa dapat di tingkatkan. Beberapa fitur tekstur yang lainnya masih bisa diuji cobakan untuk menghasilkan tingkat akurasi yang lebih baik [3].

Penelitian yang dilakukan oleh (Rismiyati, 2016) yang berjudul “Implementasi Convolutional Neural Network untuk Sortasi Salak Ekspor Berbasis Citra Digital”. Pada thesis Universitas Gadjah Mada Yogyakarta ini dikemukakan bahwa akurasi terbaik untuk model dua kelas didapatkan dengan metode CNN dengan menggunakan learning rate 0.0001, satu lapisan konvolusional dengan jumlah filter 15 dengan ukuran $3 \times 3 \times 3$, dan jumlah neuron pada lapisan tersembunyi 100. Akurasi yang didapatkan adalah 81,5%. Model 4 kelas mendapatkan akurasi 70,7% dengan 2 lapisan konvolusi [4].

Penelitian yang dilakukan oleh (L. M. Samsu, 2020) yang berjudul “Komparasi Algoritma Denoising dan Binarization dengan Adaptive Thresholding dan Morfologi untuk Meningkatkan Kualitas Keterbacaan Citra Naskah Lontar (Takepan) Sasak. penelitian yang sudah dilakukan dapat disimpulkan bahwa Adaptive Thresholding dan Morfologi dengan pendekatan dilasi sangat membantu memaksimalkan keterbacaan citra dengan nilai Peak Signal to Noise Ratio (PSNR) tertinggi mencapai 34.107 dB dibandingkan dengan teknik Denoising dan Binarization [5]

2.2. Landasan Teori

1. COVID-19

COVID-19 (*corona virus disease 2019*) pertama kali ditemukan di kota Wuhan, China pada akhir Desember 2019. Virus ini menular dengan sangat cepat dan telah menyebar ke hampir

semua negara, termasuk Indonesia, hanya dalam waktu beberapa bulan [6], [7] Coronavirus adalah kumpulan virus yang bisa menginfeksi sistem pernapasan. Pada banyak kasus, virus ini hanya menyebabkan infeksi pernapasan ringan, seperti flu. Namun, virus ini sebenarnya juga bisa menyebabkan infeksi pernapasan berat, seperti infeksi paru-paru (pneumonia). Selain virus SARS-CoV-2 atau virus Corona, virus yang juga termasuk dalam kelompok ini adalah virus penyebab *Severe Acute Respiratory Syndrome* (SARS) dan virus penyebab *Middle East Respiratory Syndrome* (MERS). Meski disebabkan oleh virus dari kelompok yang sama, yaitu coronavirus, COVID-19 memiliki beberapa perbedaan dengan SARS dan MERS, antara lain dalam hal kecepatan penyebaran dan keparahan gejala. COVID-19 lebih banyak menyerang lansia. Namun, virus ini juga bisa menyerang siapa saja, mulai dari bayi, anak-anak, hingga orang dewasa, termasuk ibu hamil dan ibu menyusui [8].

2. Kecerdasan Buatan

Kecerdasan Buatan atau Artificial Intelligence (AI) adalah teknik yang digunakan untuk meniru kecerdasan yang dimiliki oleh makhluk hidup maupun benda mati untuk menyelesaikan sebuah persoalan yang ada disekitar. Dalam penelitian ini, kecerdasan buatan yang diambil lebih dikenal dengan teknik deep learning yang

merupakan cabang dari machine learning. Menurut Ahmad ada tiga metode yang dikembangkan, antara lain:

- Machine Learning (ML) atau pembelajaran mesin merupakan teknik yang paling populer karena banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah. Sesuai namanya Machine Learning mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi.
- Fuzzy Logic (FL), teknik ini digunakan oleh mesin untuk mengadaptasi bagaimana makhluk hidup menyesuaikan kondisi dengan memberikan keputusan yang tidak kaku 0 atau 1. Sehingga dimunculkan sistem logika fuzzy yang tidak kaku. Penerapan logika fuzzy ini salah satunya adalah untuk sistem pengereman kereta api di Jepang.
- Evolutionary Computing (EC), Pendekatan ini menggunakan skema evolusi yang menggunakan jumlah individu yang banyak dan memberikan sebuah ujian untuk menyeleksi individu terbaik untuk membangkitkan generasi selanjutnya. Seleksi tersebut digunakan untuk mencari solusi dari suatu permasalahan. Contoh dari pendekatan ini adalah Algoritma Genetika yang menggunakan ide mutasi dan kawin silang, Particle Swarm Optimization (PSO) yang meniru kumpulan binatang seperti burung dan ikan

dalam mencari mangsa, Simulated Annealing yang menirukan bagaimana logam ditempa, dan masih banyak lagi [9]

3. Deep Learning

Deep learning adalah salah satu bidang machine learning yang memanfaatkan banyak layer pengolahan informasi nonlinier untuk melakukan ekstraksi fitur, pengenalan pola, dan klasifikasi [10]. Deep learning adalah sebuah pendekatan dalam penyelesaian masalah pada sistem pembelajaran komputer yang menggunakan konsep hierarki. Konsep hierarki membuat komputer mampu mempelajari konsep yang kompleks dengan menggabungkan dari konsep-konsep yang lebih sederhana. Jika digambarkan sebuah graf bagaimana konsep tersebut dibangun di atas konsep yang lain, graf ini akan dalam dengan banyak layer, hal tersebut menjadi alasan disebut sebagai deep learning (pembelajaran mendalam) [11].

4. Computer Vision

Computer vision merupakan kombinasi antara image processing dan pattern recognition. Computer vision adalah pembangunan deskripsi objek fisik yang eksplisit dan gamblang dari sebuah gambar. Output dari computer vision adalah deskripsi atau interpretasi atau beberapa pengukuran kuantitatif struktur dalam adegan 3D [10]. Computer vision adalah sebuah teknologi mesin yang mampu mengenali objek yang diamati. Kemampuan untuk mengenali ini

merupakan kombinasi dari pengolahan citra dan pengenalan pola. Pengolahan citra adalah proses awal dalam computer vision untuk menghasilkan citra yang lebih baik atau lebih mudah diinterpretasikan, sedangkan pengenalan pola adalah proses identifikasi objek pada citra. Proses-proses dalam computer vision secara garis besar dapat dibagi menjadi [12]:

- Proses mengakuisisi citra digital (Image Acquisition)
- Proses pengolahan citra (Image Processing)
- Proses analisis data citra (Image Analysis)
- Proses pemahaman data citra (Image Understanding)

2.3. Tahapan Penelitian

Adapun tahapan dari penelitian ini meliputi beberapa tahapan diantaranya :

1. Tahap awal dimulai dengan mengumpulkan dataset gambar objek penelitian
2. Langkah selanjutnya adalah dilakukan pelabelan dataset gambar untuk mempermudah penentuan letak objek yang akan dideteksi
3. Setelah proses pelabelan perlu adanya konversi berkas dari XML ke CSV untuk tujuan konversi dataset tersebut ke berkas TFRecord
4. Setelah proses konversi berkas XML dengan *output* berupa file CSV perlu adanya konversi ke *TensorFlow Record* file

yang digunakan untuk *feeding* data pada proses pelatihan

5. Kemudian dilakukan pembuatan label Map dan *dataset* (file *TFRecord*) yang sesuai dengan proses pelabelan data.
6. Kemudian langkah selanjutnya adalah konfigurasi *object detection training pipeline* untuk mengkonfigurasi proses pelatihan dan evaluasi yang dilakukan menggunakan berkas *protobuf*
7. Langkah selanjutnya adalah dilakukan proses *training* antara lain, yaitu :
 - Tahap awal pada proses pelatihan dimulai dengan *Feeding* data pelatihan atau memasukan dataset *training* ke dalam *TensorFlow*
 - Langkah selanjutnya yaitu dilakukan proses pelatihan dataset gambar untuk menghasilkan sistem pendeteksi objek Covid-19 dan Normal dengan menggunakan algoritma *Convolutional Neural Network*
 - Langkah selanjutnya adalah *Pooling Layer* yang digunakan untuk mengurangi dimensi dari *downsampling*, sehingga mempercepat komputasi karena parameter yang harus *update* semakin sedikit dan mengatasi *overfitting*
 - Kemudian dilakukan aktivasi menggunakan ReLU (*Rectrified Linear*

Units) kemudian langkah untuk proses *training* gambar

- Setelah dilakukan aktivasi, kemudian akan menghasilkan *output* prediksi kelas yang berupa gambar hasil deteksi COVID-19 dan Normal

8. Ketika *output* dari hasil pelatihan menghasilkan tingkat akurasi yang rendah maka akan dilakukan proses pelatihan kembali, akan tetapi apabila *output* menghasilkan akurasi yang tinggi akan dilanjutkan ke langkah berikutnya

9. Pada saat proses pelatihan maka akan menghasilkan *checkpoint* yang dibuat secara otomatis oleh TensorFlow berbentuk *graph tensor* yang bertujuan untuk menyimpan informasi proses pelatihan yang dilakukan, jika proses pelatihan selesai maka selanjutnya adalah mengespor *graph tensor* dan menjadikan model yang siap digunakan

10. Setelah mendapatkan model yang sesuai dengan tingkat akurasi yang tinggi dengan data yang banyak maka akan dilakukan implementasi hasil dan pembahasan

3. Metode Penelitian

3.1. Metodologi dalam Penelitian.

Metode Penelitian menggunakan studi pustaka yaitu dengan mencari referensi tentang algoritma *Convolutional Neural Network (CNN)* yang

bertujuan untuk mendeteksi dan mengklasifikasi objek COVID-19 dan Normal berdasarkan dataset gambar hasil x-ray rontgen foto thorax yang diperoleh dari situs Kaggle [18].

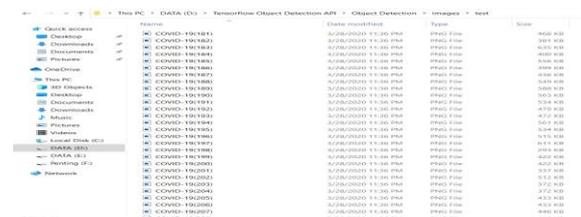
3.2. Lokasi Penelitian

Untuk kebutuhan Dataset yang digunakan diambil sample hasil ronsen beberapa pasien di Rumah Sakit Umum Soejono Selong Lombok Timur

4. Hasil dan Pembahasan

4.1. Persiapkan Dataset

Dalam studi kasus ini, digunakan dataset gambar terinfeksi COVID-19 dan Normal dengan masing-masing sebanyak 358 data training dan 78 data testing, sehingga total dataset sebanyak 436 images berbentuk file PNG. File ini masing-masing dimasukkan ke folder image / test dan image /train.



Gambar 1. Dataset PNG Image/test

Pelabelan Gambar kedalam Format .xml

Pemberian label pada gambar dataset dapat disebut dengan *annotations*. *Annotations* ini nantinya akan disimpan dalam *file* berformat .xml. Aplikasi yang digunakan dalam pelabelan adalah Labellmg.

Berikut langkah-langkah dalam pelabelan :

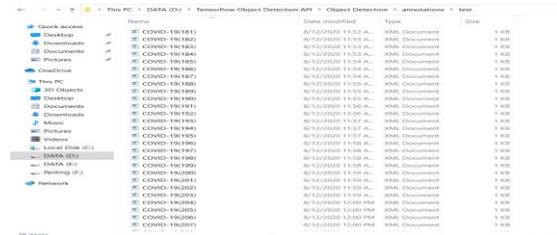
- Buka aplikasi labellmg.

- Klik tombol “OpenDir”
- Pilih direktori *dataset* gambar
- Klik tombol “Change Save Dir” untuk menyimpan hasil file pelabelan .xml ke dalam 1 direktori yang sama (Annotation)
- Klik tombol “Create RectBox” untuk membuat kotak area objek yang akan dikenali (klik W)
- Arahkan kursor dan tarik area kotak disekitar objek
- Lalu akan muncul kotak dialog untuk memberikan nama label dari objek yang akan kita kenali (Nama objek setiap kategori harus sama).
- Simpan hasil pelabelan dengan menekan tombol CTRL+S (klik *Change Save Dir*)
- Simpan *file* hasil pelabelan *.xml kedalam folder *annotations*

Dalam studi kasus ini, digunakan dataset gambar terinfeksi COVID-19 dan Normal dengan masing-masing sebanyak 358 data *training* dan 78 data *testing*, sehingga total dataset *.xml terbentuk juga sebanyak 436 *images*.

File ini masing-masing dimasukkan kedalam folder *annotation/test* dan *annotations/train*.

Berikut adalah hasil yang sudah di label ke format .xml yang di simpan pada *folder annotation/test*:

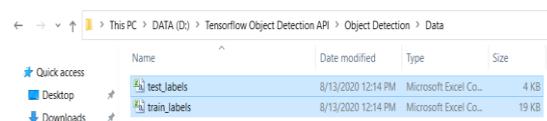


Gambar 2. Hasil pelabelan dengan format .xml

4.2. Convert file .xml ke .csv

Di direktori yang telah dibuat sebelumnya terdapat file dengan nama “xml_to_csv” yang berisi kode dalam mengkonversi file xml ke csv.

Jalankan perintah berikut untuk melakukan konversi xml ke csv pada command promt: python xml_to_csv.py



Gambar 3. Hasil convert .xml ke .csv

4.3. Convert .csv ke TFRecord

Pada saat melakukan proses training, tensorflow akan membaca data input dalam format TFRecord yang dinamakan feeding data. Oleh karena itu perlu dilakukan generate data annotation yang tadi telah dikonversi ke file .csv. Untuk kode TFRecord sendiri juga sudah ada dengan nama file “generate_tf_record”. Hal yang perlu dilakukan adalah merubah kelas sesuai dengan kelas yang akan didefinisikan. Karena pada kasus memiliki 2 objek yang akan dideteksi maka kelas yang diubah seperti berikut:

```

33 # TO-DO replace this with label map
34 def class_text_to_int(row_label):
35     if row_label == 'normal':
36         return 1
37     elif row_label == 'covid 19':
38         return 2
39     else:
40         return 0
41

```

Gambar 4. Setting kelas sesuai objek penelitian

4.4. Konfigurasi Pipeline

Konfigurasi pipeline disini berhubungan untuk mengatur file config yang nantinya akan digunakan untuk melakukan konfigurasi dari model training karena *tensorflow* menggunakan *protobuf* maka konfigurasi ini sangat diperlukan. Model yang digunakan adalah *Faster-RCNN-Inception-V2* dimana model ini sudah disediakan oleh *Tensorflow* itu sendiri. Untuk konfigurasi yang harus dilakukan terdapat pada file *faster_rcnn_inception_v2_pets* yang selanjutnya kalian simpan di folder training.

4.5. Training

Setelah semua data yang dibutuhkan untuk training sudah siap. Maka untuk menjalankan training ketik perintah berikut pada command prompt : `python train.py --logtostderr --train_dir = training/ pipeline _ config _ path = training/faster _ rcnn _ inception_v2_pets.config`. Jika berhasil maka akan muncul proses training seperti dibawah ini:

```

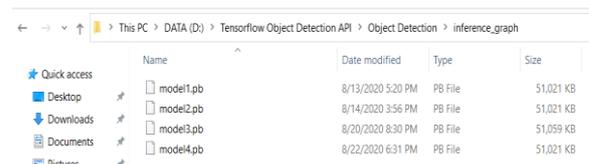
C:\WINDOWS\system32\cmd.exe - python train.py --logtostderr --train_dir=training/ --pipeline_config_path=training/faster_rcnn_inception_v2_pets.config --device=GPU --gpu_device_id=0 --device_id=0 --max_queue_size=500 --ps_isolation_strategy=LOCAL --compute_capability=6.1
INFO:tensorflow:Restoring parameters from C:/tensorflow/models/research/object_detection/faster_rcnn_inception_v2_coco_2018_01_28/model.ckpt
INFO:tensorflow:Starting Session.
INFO:tensorflow:Saving checkpoint to path training/model.ckpt
INFO:tensorflow:Starting Queue.
INFO:tensorflow:global_step/sec: 0
INFO:tensorflow:global_step: 0
INFO:tensorflow:global_step: 1: loss = 2.0788 (1.383 sec/step)
INFO:tensorflow:global_step: 2: loss = 1.8832 (0.251 sec/step)
INFO:tensorflow:global_step: 3: loss = 2.4888 (0.298 sec/step)
INFO:tensorflow:global_step: 4: loss = 2.9733 (0.193 sec/step)
INFO:tensorflow:global_step: 5: loss = 2.2184 (0.191 sec/step)
INFO:tensorflow:global_step: 6: loss = 2.8121 (0.554 sec/step)
INFO:tensorflow:global_step: 7: loss = 2.8454 (0.254 sec/step)
INFO:tensorflow:global_step: 8: loss = 2.8252 (0.288 sec/step)
INFO:tensorflow:global_step: 9: loss = 2.8851 (0.258 sec/step)
INFO:tensorflow:global_step: 10: loss = 1.3422 (0.193 sec/step)
INFO:tensorflow:global_step: 11: loss = 1.8827 (0.197 sec/step)
INFO:tensorflow:global_step: 12: loss = 1.2485 (0.196 sec/step)
INFO:tensorflow:global_step: 13: loss = 1.8712 (0.193 sec/step)
INFO:tensorflow:global_step: 14: loss = 1.6686 (0.189 sec/step)
INFO:tensorflow:global_step: 15: loss = 1.2657 (0.192 sec/step)
INFO:tensorflow:global_step: 16: loss = 1.4103 (0.193 sec/step)
INFO:tensorflow:global_step: 17: loss = 1.2152 (0.192 sec/step)
INFO:tensorflow:global_step: 18: loss = 1.1165 (0.192 sec/step)
INFO:tensorflow:global_step: 19: loss = 1.6557 (0.192 sec/step)
INFO:tensorflow:global_step: 20: loss = 1.7777 (0.208 sec/step)

```

Gambar 5. Training menggunakan Tensorflow
Selain menggunakan proses training pada komputer, peneliti juga bisa menggunakan RAM dan GPU gratis milik google, yaitu Google Colaboratory.

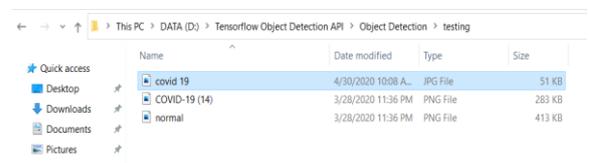
4.6. Export Inference Graph

Untuk melakukan *export inference graph* dapat dilakukan dengan menulis perintah berikut pada command prompt : `python export_inference_graph.py --input_type image --tensor_pipeline_config_path training/faster_rcnn_inception_v2_pets.config --trained_checkpoint_prefix training/model.ckpt -XXXX --output_directory inference_graph` model.ckpt -XXXX dapat diganti dengan nilai step yang akan di export. Nilai ini dapat dilihat pada folder training, misal: ketika kita setting jumlah step 25.000 maka nilai XXX tersebut diganti dengan nilai 25.000. Hasil export model tersebut dapat dilihat pada folder *inference_graph*:



Gambar 6. Save model

4.7. Testing Image



Gambar 7. Testing Data Baru

Setelah mendapatkan *frozen _ inference _ graph.pb* untuk melakukan testing image, dapat dilakukan dengan setting beberapa coding di file *Object_detection_image.py*. Setting yang dilakukan diantaranya, mengganti image yang akan di test pada line 39, mengganti jumlah *NUM_CLASSES* dengan jumlah kelas yang dimiliki yaitu 2, *frozen_inference_graph.pb* diubah menjadi *model4.pb*, serta mengganti *IMAGE_NAME* sesuai data yang ingin di uji coba.

```
15 # Import utilities
16 from utils import label_map_util
17 from utils import visualization_utils as vis_util
18
19 # Name of the directory containing the object detection module we're using
20 MODEL_NAME = 'inference_graph'
21 IMAGE_NAME = 'testing/covid 19.jpg'
22
23 # Grab path to current working directory
24 CWD_PATH = os.getcwd()
25
26 # Path to frozen detection graph .pb file, which contains the model that is used
27 # for object detection.
28 PATH_TO_CKPT = os.path.join(CWD_PATH, MODEL_NAME, 'model4.pb')
29
30 # Path to label map file
31 PATH_TO_LABELS = os.path.join(CWD_PATH, 'data', 'label_map.pbtxt')
```

Gambar 8. Setting Script Object Detection

Jika sudah selesai maka untuk melakukan uji coba dapat dilakukan dengan menjalankan code python *Object _ detection _ image.py* pada command prompt, sehingga ketika proses setting tersebut selesai, maka dapat dilihat hasil uji coba dari data tersebut dengan ditandai kotak berwarna kuning untuk data gambar COVID-19. Berikut adalah persentase akurasi dari data diatas, yaitu:



Gambar 9. Hasil Deteksi COVID-19 pada model4

Berdasarkan uji coba *image detector* pada Gambar 17 diperoleh hasil pendeteksian *x-ray rontgen* terinfeksi COVID-19, yaitu dengan nilai akurasi sebesar 98% pada model4 dengan proses training 25.000 step dan 1.000 *evaluation_step*.

5. Kesimpulan

Dari hasil analisis dan pembahasan dalam studi kasus pada tugas akhir ini, maka dapat diambil beberapa simpulan sebagai berikut:

Dengan menggunakan 25.000 step dan 1.000 *evaluation_step* untuk melakukan pengujian model4 pada data baru terinfeksi COVID-19 menghasilkan akurasi sebesar 98%.

Tingkat akurasi pendeteksian thorax terinfeksi COVID-19 pada model4 menggunakan algoritma Convolutional Neural Network dapat dinilai bekerja dengan baik dengan beragam tingkat akurasi pada hasil pengujian berkisar 60-99%.

6. Daftar Pustaka

- [1] W. Moh. Farid and S. Jagat, "Pemanfaatan Teknik Pengenalan Wajah Berbasis Opencv untuk Sistem Informasi Pencatatan Kehadiran Dosen," *Infotek: Jurnal Informatika dan Teknologi*, vol. 1, no. 2, pp. 96 - 106, Juli 2018.
- [2] I. Fathurrahman and I. Gunawan, "Pengenalan Citra Logo Kendaraan Menggunakan Metode Gray Level Co-Occurrence Matrix (GlcM) dan Jst-Backpropagation," *Infotek: Jurnal*

- Informatika dan Teknologi*, vol. 1, no. 1, p. 47 – 55, Januari 2018.
- [3] I. Fathurrahman, A. Muliawan Nur and Fathurrahman, "Identifikasi Kematangan Buah Mentimun Berbasis Citra Digital Menggunakan Jaringan Syaraf Tiruan Backpropagation," *Infotek: Jurnal Informatika dan Teknologi*, vol. 1, no. 2, pp. 27 - 33, Januari 2019.
- [4] Rismiyati, "Implementasi Convolutional Neural Network untuk Sortasi Mutu Salak Ekspor Berbasis Citra Digital," Yogyakarta, 2016.
- [5] M. S. L.M. Samsu, "Komparasi Algoritma Denoising Dan Binarization Dengan Adaptive Thresholding Dan Morfologi Untuk Meningkatkan Kualitas Keterbacaan Citra Naskah Lontar (Takepan) Sasak," *Jurnal Informatika dan Teknologi*, vol. 3, pp. 204-210, 02 07 2020.
- [6] Huang and e. al, "Clinical features of patients infected with 2019 novel coronavirus in Wuhan China," *The Lancet*, vol. 6736(20), pp. 1-10, 2020.
- [7] M. Saiful, "Implementasi Algoritma Naive Bayes Untuk Memprediksi Predikat Ketuntasan Belajar Siswa Pasca Pandemi Covid 19," *Jurnal Informatika*, vol. 4, pp. 29-38, 2021.
- [8] D. Wang and e. al, "Clinical Characteristics Of 138 Hospitalized Patients With 2019 Novel Coronavirus-Infected Pneumonia in Wuhan, China," *Journal of The American Medical Association*, vol. 323(11), pp. 1061-1069, 2020.
- [9] A. Ahmad, "Mengenal Artificial Intelligence, Machine Learning, Neural Network, dan Deep Learning," *Jurnal Teknologi Indonesia*, 2017.
- [10] L. & Y. D. Deng, *Deep Learning: Methods and Application, Foundations and Trends in Signal Processing.*, 2014. .
- [11] LeCun, Y. Bengio and G. Y. Hinton, "Deep Learning," Retrieved from *Nature International Journal of Science*, vol. 521 (7533), pp. 436-444, 2015..
- [12] L. F. Basuki, 05 2020. [Online]. Available: <http://elib.unikom.ac.id/gdl.php?mod=browse&op=read &id=jbptunikompp-gdl-lutfifebri-35958>.
- [13] F. Jalled, Object Detection Using Image Processing, Diakses dari <https://arxiv.org/pdf/1611.07791.pdf>, 2016..
- [14] Goodfellow, I. Bengio and A. Y. Courville, *Deep Learning (Adaptive Computation and Machine Learning Series)*, The MIT Press, 2016.
- [15] Ldya and e. al, "Pengertian Citra," Universitas Sumatera Utara, Medan, 2010.
- [16] R. Nouroz, "What is the benefit of using average pooling rather than max pooling," 05 2020. [Online]. Available: <https://www.quora.com/What-is-the-benefit-of-using-average-pooling-rather-than-max-pooling..>
- [17] S. Sagar, "Activation Functions: Neural Networks," 05 2020. [Online]. Available: <https://towardsdatascience.com/activation-functions-neural-networks1cbd9f8d91d6>.
- [18] T. Rahman, Maret 2020. [Online]. Available: <https://www.kaggle.com/tawsifurrahman/covid19-radiography-database?select=COVID-19+Radiography+Database>.
- [19] A. Sudianto and J. Sugiantara, "Website as Foundation Information Media under the auspices of Nahdlatul Wathan," *J. Phys. Conf. Ser.*, vol. 1539, no. 1, pp. 3–8, 2020, doi: 10.1088/1742-6596/1539/1/012024