SATIN – Sains dan Teknologi Informasi
Journal Homepage : http://jurnal.sar.ac.id/index.php/satin

Implementasi Machine Learning Sebagai Pengenal Nominal Uang Rupiah dengan Metode YOLOv3

Aditiya Hermawan [1], Leonardo Lianata [2], Ardiane Rossi Kurniawan Maranto [3], Junaedi[4]
[1]Universitas Buddhi Dharma, aditiya.hermawan@ubd.ac.id, Rawa Kompeni, Tangerang, Indonesia
[2]Universitas Buddhi Dharma, leonardo.lianata@ubd.ac.id, Karawaci, Tangerang, Indonesia
[3]Universitas Buddhi Dharma, ardiane.rossi@ubd.ac.id, Setu, Tangerang Selatan, Indonesia
[4]Universitas Buddhi Dharma, junaedi@ubd.ac.id, Desa Jatimulya, Tangerang, Indonesia

Informasi Makalah

Submit : Mar 23, 2022
Revisi : Apr 8, 2022
Diterima : Jun 7, 2022

Kata Kunci :

Machine Learning
YOLOv3
Android

Abstrak

Jumlah disabilitas kesulitan melihat (Tunanetra) di atas 10 tahun sebanyak 6,36% dari total penduduk yang mengalami disabilitas yaitu 8,56% pada tahun 2015. Permasalahan yang dihadapi penyandang tunanetra dalam kehidupan sehari-hari salah satunya mengenali nominal uang rupiah. Walaupun pemerintah sudah membuat uang dengan emboss pada emisi 2016, tetapi masih kurang efektif karena uang yang beredar kadang dalam kondisi tidak rapih. Untuk mengatasi hal tersebut dapat dibantu dengan menggunakan teknologi Machine learning berbasis Yolov3 dalam mengenali nominal uang Rupiah. Metode YOLOv3 mempunyai keunggulan dalam kecepatan pelatihan model dan nilai akurasinya yang tinggi, dan memang dirancang untuk mengolah gambar. Dataset yang digunakan untuk membuat model machine learning dikumpulkan dari berbagai gambar uang rupiah nominal 1000, 2000, 5000, 10000, 20000, 50000, 10000 sebanyak 4200 gambar. Model yang sudah dibuat selanjutnya diimplementasikan kedalam bentuk aplikasi android. Aplikasi dijalankan seperti melakukan scan uang dan memberikan hasil berupa suara yang menyebutkan nominal uang tersebut secara otomatis. Model ini dievaluasi dengan Confusion Matrix menghasilkan nilai accuracy, precision dan recall sebesar 0.98. Berdasarkan Nilai akurasi tersebut, model yang dibuat dapat membantu penyandang tunanetra dalam mengenali nominal uang rupiah.

Abstract

The number of visually impaired (blind) over 10 years is 6.36% of the total population with disabilities, namely 8.56% in 2015. One of the problems faced by blind people in daily life is recognizing the nominal rupiah currency. Even though the government has made money by embossed on 2016 emissions, it is still less effective because the money in circulation is sometimes in an untidy condition.To overcome this, it can be helped by using machine learning technology based on Yolov3 in recognizing the nominal value of Rupiah. The YOLOv3 method has advantages in the speed of model training and high accuracy values, and is indeed designed

Aditiya Hermawan
Email: aditiya.hermawan@ubd.ac.id.

to process images. The dataset used to create the machine learning model was collected from various images of IDR 1000, 2000, 5000, 10000, 20000, 5000, 10000 as many as 4200 images. The model that has been made is then implemented in the form of an android application. The application is run like scanning money and gives the result in the form of a voice stating the nominal money automatically. This model is evaluated by Confusion Matrix resulting in accuracy, precision and recall values of 0.98. Based on the accuracy value, the model created can help blind people in recognizing the nominal rupiah currency.

1. Pendahuluan

Sebanyak 8,56% penduduk yang berumur diatas 10 Tahun yang mengalami gangungan fungional menurut survei dari penduduk antar sensus tahun 2015, (Informasi Kementrian Kesehatan RI, 2019). Tunanetra adalah suatu istilah yang digunakan bagi orang yang mengalami gangguan atau hambatan dalam indera penglihatan baik secara total maupun sebagian. Tetapi kondisi ini tidak berarti bahwa penyandang tunanetra tidak dapat melakukan aktivitas sehari-hari seperti melakukan transaksi yang mengunakan uang rupiah. Bank Indonesia selaku pihak yang mengeluarkan uang rupiah seri Pahlawan Nasional emisi 2016 juga menyertakan *emboss* atau permukaan yang timbul pada sisi-sisi uang agar tunanetra dapat membedakan nilai uang. Tetapi pada kenyataannya banyak sekali uang yang beredar sudah tidak lagi rapih hingga banyak tekukan. Hal ini akan membuat penyandang tunanetra kesulitan dalam mengidentifikasi nominal uang, karena tekukan-tekukan yang ada dapat menyamarkan *emboss* pada uang.

Solusi untuk membantu penyandang tunanetra di ataranya dengan mengunakan teknologi yang dapat mengidentifikasi jumlah atau nominal uang rupiah secara langsung dan memiliki output suara yang menyebutkan jumlah atau nominal uang tersebut. Hal tersebut dapat dilakukan dengan mengimplementasikan Machine Learning. *Machine Learning* diartikan sebagai proses pembelajaran terhadap mesin dari data-data yang kita berikan sehingga mesin tersebut dapat berpikiran dan berperilaku layaknya manusia. Pendekatan *Machine Learning* dilakukan dengan cara pelatihan terhadap mesin dengan tujuan khusus agar dapat memprediksi hasil berdasarkan dataset yang ada (Simeone, 2018). Metode *Machine Learning* cocok untuk diterapkan pada permasalahan pencocokan gambar, karena dapat membuat sebuah komputer atau program yang mempunyai kecerdasan seperti manusia. Machine Learning dapat melakukan pendugaan pada sebuah gambar yang tidak harus sama dengan sumbernya dari kondisi gambar, cahaya bahkan warna yang pudar karena model yang dibuat tidak hanya bersumber pada 1 gambar saja, tapi dari beberapa data yang sudah di pelajari pada tahap pemodelan. Sebelum melakukan prosedur terkait data, maka penting untuk membersihkan dan memformat data karena umumnya raw data mengandung noise, nilai yang hilang, dan format yang tidak sesuai yang tidak dapat digunakan untuk model *machine learning* secara langsung (Sianturi, J., & Hajjah, 2021).

Cakupan *Machine Learning* sangatlah luas, namun ada juga beberapa bagian ilmu yang sudah dibagi untuk mempelajari bidang-bidang yang ada didalam *Machine Learning* secara spesifik seperti Deep Learning. *Deep Learning* merupakan ekstrak tingkat tinggi, abstraksi kompleks sebagai representasi data melalui sebuah proses pembelajaran hierarkis (Najafabadi et al., 2015). Hal ini bisa dilihat dari berbagai ragam riset yang sudah dilakukan, sebagian memfokuskan pada implementasi dan penyelarasan model machine learning (Yuni Eka Achyani, 2018)(Anjar Wanto, 2018).

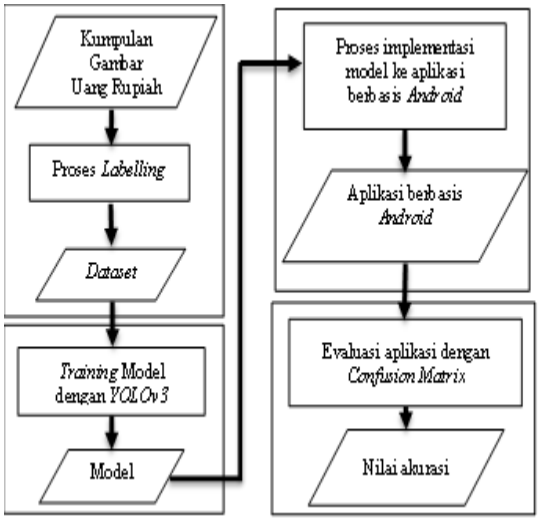
SATIN – Sains dan Teknologi Informasi
Vol. 8, No. 1, Juni 2022, pp. 12-22
ISSN: 2527-9114, DOI: 10.33372/stn.v8i1.816

Terdapat beberapa bagian dari deep learning, salah satunya adalah Jaringan saraf tiruan atau Neural Network. Algoritma Neural Network pada dasarnya merupakan kumpulan pasangan input-output yang saling terkoneksi (Hadiwandra, 2019). Neural Network merupakan kumpulan elemen, unit atau node pemrosesan yang saling berhubungan. Kemampuan pemrosesan jaringan disimpan dalam kekuatan koneksi interunit atau bobot yang diperoleh dengan proses adaptasi atau belajar dari serangkaian pola pelatihan (Cross et al., 1995).

Salah satu metode yang menerapkan *Neural Network* adalah *YOLO* atau *You Only Look Once* versi 3, *YOLO* sangat cepat, karena membingkai deteksi sebagai permasalahan regresi yang tidak membutuhkan *complex pipeline*. Dengan menjalankan *Neural Network* pada gambar baru pada waktu pengujian untuk memprediksi deteksi (Redmon et al., 2016). Terdapat beberapa versi dari YOLO diantaranya adalah YOLOv3 yang biasanya digunakan untuk memecahkan masalah dalam ruang lingkup jaringan saraf *multi-layer* atau yang berhubungan dengan gambar, salah satunya untuk mengenali gambar atau objek. *YOLOv3* juga mempunyai keunggulan dalam proses pelatihannya yang cepat dan menghasilkan akurasi yang tinggi. Pada penelitian ini menggunakan *YOLOv3* untuk melakukan deteksi objek nomilan uang rupiah, dengan metode *YOLOv3* sangat memungkinkan dalam memberikan hasil lebih cepat dan maksimal dibandingkan dengan metode lain yang serupa.

2. Metode Penelitian

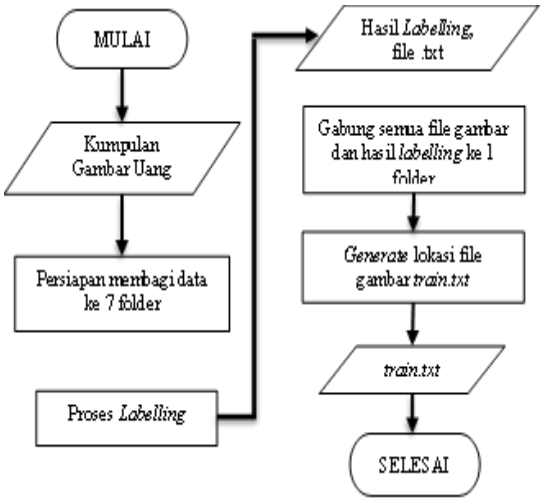
Metode yang digunakan pada penelitian ini terdapat 4 tahap yaitu 1. Perancangan Dataset, 2. Melatih Model Machine Learning, 3. Implemtasi, 4. Evaluasi Modsel.



Gambar 1. Tahapan metode penelitian.

A. Perancangan Dataset

Perancangan model dimulai dengan melakukan perancangan dataset yang mana terdapat 2 tahap utama yaitu Pengumpulan dan Persiapan Data. Pengumpulan data gambar dengan cara membagi ke 7 folder sesuai dengan nominalnya. Selanjutnya tahap *labelling* yang nantinya akan menghasilkan file *namafile.txt*. Setelah *labelling*, gabung kembali semua file gambar berserta .txt nya ke satu folder, lalu melakukan *generate* lokasi file gambar sehingga menghasilkan file *train.txt*.



Gambar 2. Flowchart perancangan dataset.

SATIN – Sains dan Teknologi Informasi
Vol. 8, No. 1, Juni 2022, pp. 12-22
ISSN: 2527-9114, DOI: 10.33372/stn.v8i1.816

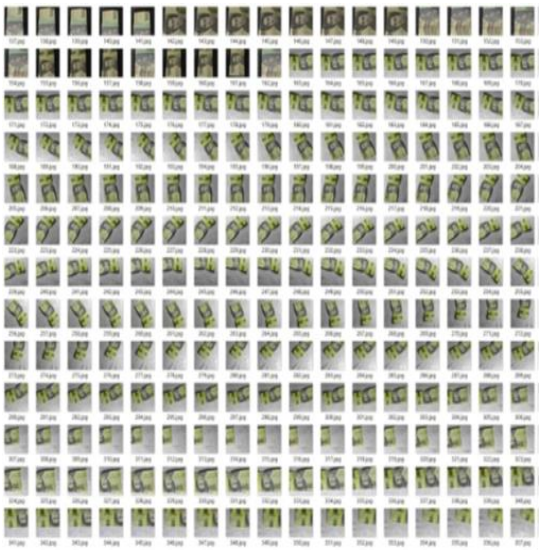
1. Pengumpulan Data

Data yang digunakan untuk membuat model adalah data Primer yang diambil langsung dengan cara melakukan pengumpulan gambar dengan menggunakan kamera. Gambar yang digunakan adalah uang kertas sesuai dengan sejumlah 7 jenis dengan nominal yang berbeda. Proses pengambilan gambar dilakukan menggunakan kamera *smartphone* dengan *burst mode*, lalu gambar dipindahkan ke komputer. Setiap nominal uang terdiri dari 600 gambar uang yang bervariasi, gambar tampak depan dan belakang uang, *landscpace*, *portrait*, gambar wajah dan sisi-sisi uang. Semua gambar berformat .*JPG*.

2. Persiapan Data

Setelah semua gambar dikumpulkan. Selanjutnya dilakukan pengelompokan atau membagi data ke dalam folder sesuai dengan nominalnya masing-masing. Dalam tahap ini, peneliti membagi data kedalam 7 folder yang masing-masing folder bernama 1000, 2000, 5000, 10000, 20000, 50000, 10000 yang bertujuan untuk mempermudah proses *labelling*.

Gambar 3. Pengelompokan gambar.

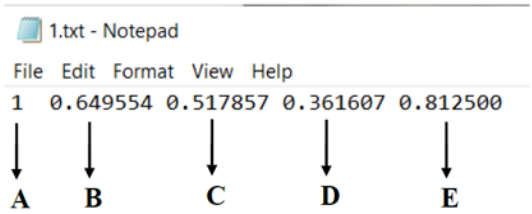


Gambar 4. Contoh dataset gambar.

3. Labelling

Proses *labelling* dilakukan dengan *labeling*, dimana prosesnya akan menghasilkan file anotasi *YOLOv3* berbentuk *.txt* yang diberi nama difilenya sama dengan nama gambarnya. Proses ini dilakukan terhadap semua gambar secara satu per satu.

Gambar 5. Proses *labelling*.



Gambar 6. Hasil label dalam bentuk .txt.

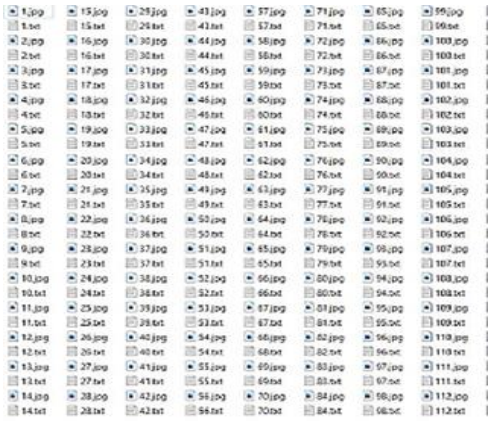
Hasil proses *labelling* dalam bentuk .txt seperti Gambar 4 berisi informasi dari gambar yang nantinya digunakan dalam proses pelatihan model, dimana format anotasinya terdiri atas:

A = *object-class* (kelas atau class)
B = *x_center* (lebar dari titik pusat)
C = *y_center* (tinggi dari titik pusat)
D = *width* (lebar gambar)
E = *height* (tinggi gambar)

4. Gabungkan File Gambar

Setelah selesai melakukan proses labelling untuk semua gambar yang ada, kemudian semua file di jadikan dalam 1 folder.

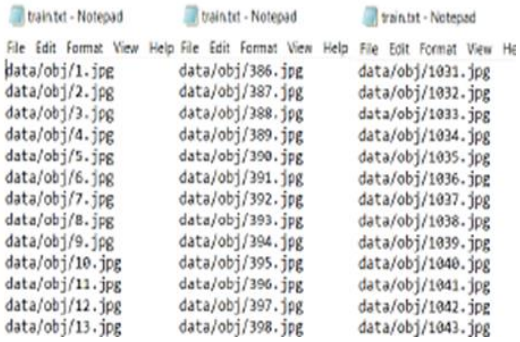
SATIN – Sains dan Teknologi Informasi
Vol. 8, No. 1, Juni 2022, pp. 12-22
ISSN: 2527-9114, DOI: 10.33372/stn.v8i1.816



Gambar 7. Hasil penggabungan file gambar dan *.txt* ke satu folder.

5. Generate Lokasi File

Pada tahap ini, dilakukan proses generate lokasi file gambar dalam folder *train* menjadi :



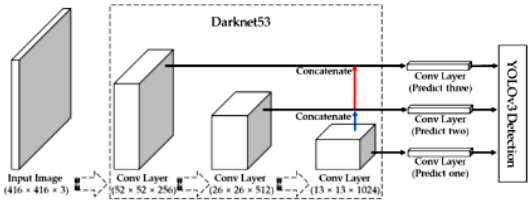
Gambar 8. Isi *file train.txt*.

B. Model *Machine Learning*

Machine Learning adalah sebuah proses pembelajaran terhadap mesin dari data-data yang kita berikan sehingga mesin tersebut dapat berpikiran dan berperilaku seperti manusia (Simeone, 2018). Tipe algoritma *Machine Learning* diantaranya adalah *Supervised Learning, Unsupervised Learning, Semi-Supervised Learning, Reinforcement Learning, Multi Task Learning, Ensemble Learning, Neural Netowrk, Instant Base Learning* (Dey, 2016). *Machine Learning* merupakan bagian dari kecerdasan buatan yang terfokus pada suatu aplikasi yang belajar sendiri dari data yang dilatih dengan algoritma yang efisien untuk memecahkan suatu masalah, data yang dimaksud dapat berupa angka, kata, gambar, dan lain-lain yang nantinya akan dimasukan kedalam suatu algoritma *machine learning* (Andre Kanisius Edguard Lapian, 2021).

Membuat dan melatih model berdasarkan dataset yang telah dilabel yang telah dibuat, yang nantinya model akan digunakan untuk mengenali nilai atau nominal mata uang. Pelatihan model menggunakan *YOLOv3* (*You Only Look Once*).

YOLO (You Only Look Once) merupakan sebuah jaringan konvolusional tunggal yang secara bersamaan memprediksi beberapa kotak terikat dan probabilitas kelas untuk kotak-kotak tersebut. YOLO melatih gambar penuh dan secara langsung mengoptimalkan kinerja deteksi. Model ini memiliki beberapa manfaat dibandingkan metode tradisional deteksi objek. YOLO sangat cepat. Karena membingkai deteksi sebagai masalah regresi, YOLO tidak memerlukan pipeline yang rumit. YOLO hanya menjalankan neural network pada gambar baru pada waktu uji (Redmon et al., 2016).



Gambar 9. Struktur YOLOv3 (Ma et al., 2020)

Untuk *framework machine learning* yang dipakai dibangun secara *opensource* dengan dukungan dari perusahaan raksasa teknologi *Google* yaitu *TensorFlow* (Adie; Renata Tresy, 2018). *TensorFlow* menggunakan grafik aliran data untuk merepresentasikan komputasi, status data, dan operasi yang mengubah status tersebut. *TensorFlow* mendukung pelatihan dan inferensi dalam skala besar dengan efisien menggunakan banyak *server* yang berkemampuan tinggi untuk melakukan pelatihan dengan cepat, dan menjalankan model terlatih untuk inferensi dalam produksi di berbagai *platform*, mulai dari cluster terdistribusi di pusat data, hingga

SATIN – Sains dan Teknologi Informasi
Vol. 8, No. 1, Juni 2022, pp. 12-22
ISSN: 2527-9114, DOI: 10.33372/stn.v8i1.816

berjalan secara lokal pada perangkat seluler (Abadi et al., 2016).

C. Implementasi

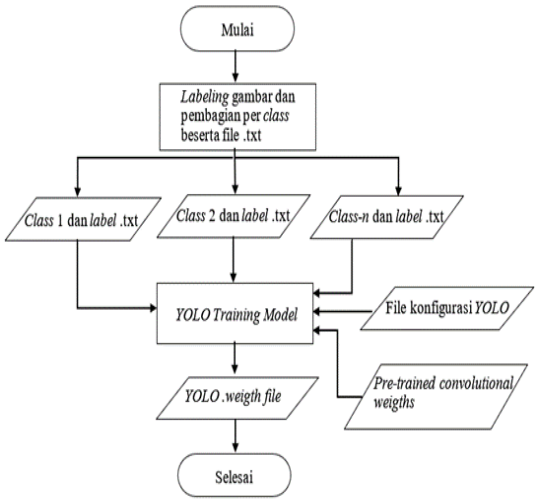
Hasil dari model yang dibentuk kemudian di implementasikan dalam kedalam aplikasi berbasis android.

D. Evaluasi Model

Pengujian dan evaluasi pada model dan aplikasi yang telah dibuat dilakukan dengan menggunakan Confusion Matrix untuk mengetahui nilai akurasi dari aplikasi.

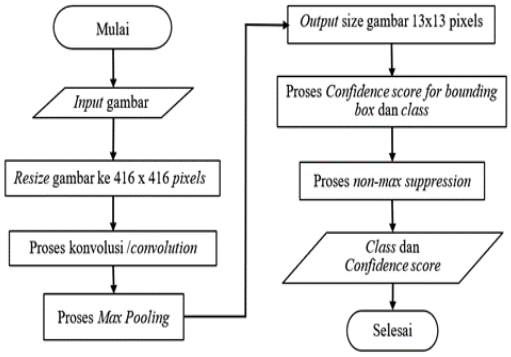
3. Hasil dan Pembahasan

Proses pelatihan menggunakan metode *YOLOv3* dimulai dengan menyiapkan *dataset* yang telah di label yang memiliki file .txt. Membagi file gambar dan *label* .txt ke masing-masing folder sesuai kelasnya. Selanjutnya melakukan konfigurasi file *config YOLOv3* sesuai dengan *dataset*. Menyiapkan file *pretrained convolutional weigths*. Kemudian memulai proses training model dengan *YOLOv3*, sehingga akhirnya menghasilkan file model yang telah dilatih dengan format .weigths.



Gambar 10. *Flowchart* pelatihan *YOLOv3* (Shinde et al., 2018).

Proses pengenalan gambar dengan *model YOLOv3* dimulai dengan *input* gambar. Gambar yang *input* di *resize* ke 416 x 416 *pixels*. Seletah di-*resize* masuk ke proses konvolusi dan *max pooling* sehingga *size* gambar menjadi 13x13 *pixels*. Dari 13x13 *pixels* dilakukan semua probabilitas *box* beserta *class* dan *confidence score*-nya. Dilanjutkan dengan proses *non-max suppression*, dimana mengeliminasi box yang *confidence score*-nya rendah. Maka mendapatkan hasil akhir *class* beserta *confidence score*-nya.



Gambar 11. *Flowchart* proses pengenalan gambar dengan *mode YOLOv3*.

A. Persiapan Training Model

Persiapan *training* terhadap *dataset* yang telah dibuat, dimulai dengan melakukan konfigurasi untuk proses *training* model *YOLOv3*, lalu membuat beberapa file pendukung seperti file yang berisi nama atau *label* objek dan file konfigurasi yang berisi lokasi file pendukung serta lokasi *output* model.

Konfigurasi untuk model yang di latih :

max_batches : 14000 didapatkan dari rumus (jumlah class*2000), dikarenakan model ini ada 7 *class*, maka *max_batches* adalah 7*2000 = 14000.

Step : 11200,12600 didapatkan dari rumus (80% , 90%

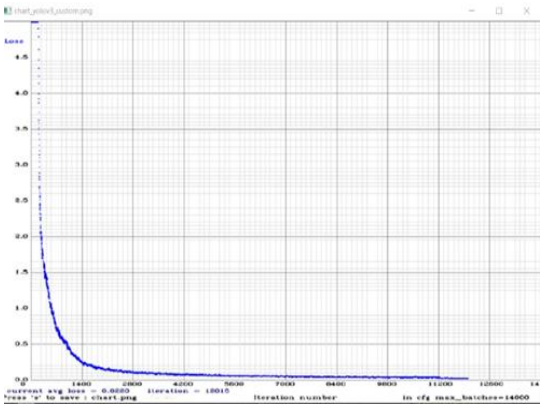
SATIN – Sains dan Teknologi Informasi
Vol. 8, No. 1, Juni 2022, pp. 12-22
ISSSN: 2527-9114, DOI: 10.33372/stn.v8i1.816

max_batches), jadi 80% dari 14000 = 11200 dan 90% dari 14000 = 12600.

Classes : 7 , jumlah *class* pada model yang terdiri dari (seribu, duaribu, limaribu, sepuluhribu, duapuluhribu, limapuluhribu, seratusribu).

B. Pelatihan Model

Proses *training* model dilakukan sebanyak 14000 iterasi, sesuai dengan konfigurasi pada *YOLOv3* pada tahap sebelumnya dan proses berjalan otomatis. Berikut adalah hasil proses pelatihan model:

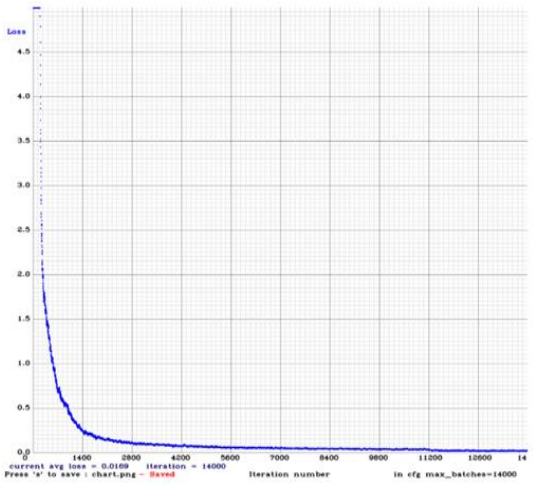


Gambar 12. Proses pelatihan model

C. Hasil Training Model

Proses *Training* model menghasilkan 16 file model dimana setiap 1000 iterasi dihasilkan 1 *file* model atau *break-point* nya. Model yang dihasilkan berformat *.weights*. File model yang penulis gunakan untuk penelitian adalah *file* terakhir atau yang bernama *yolov3_custom_last.weights* Dari hasil *Training* menghasilkan model terakhir dengan total *avg_loss* sebesar 0,0169 atau 1,69% yang berarti model mempunyai tingkat akurasi 98,31%.

Nilai *avg_loss* adalah nilai rata-rata terjadinya kesalahan dalam mengenal objek. Nilai tersebut didapatkan dari hasil perhitungan rumus dibawah, dimana *Loss* dari *YOLOv3* tersusun dari 3 bagian, yaitu *co-ordinate prediction error*, *confidence score* dan *intersection over union* atau *IoU*.



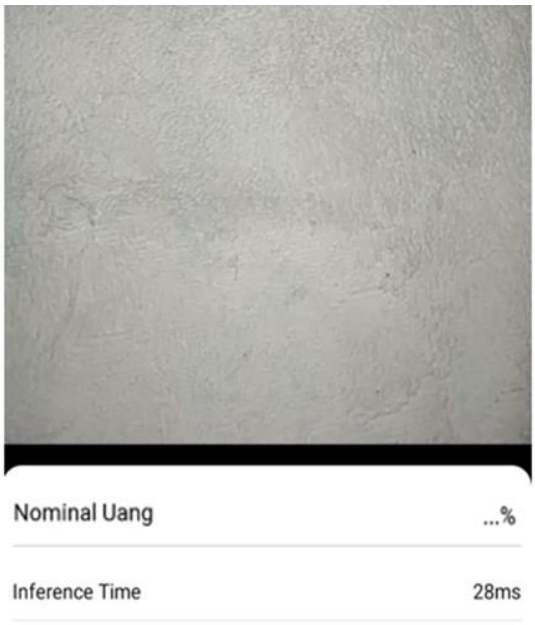
Gambar 13. Grafik *avg_loss training* model

Model yang digunakan adalah *yolov3_custom_last.weights* akan dikonversi formatnya menjadi bentuk *.tflite* atau *Tensorflow Lite* agar dapat diimplementasikan ke dalam tahap selanjutnya.

D. Implementasi Aplikasi Android

Sejak Android dirilis pada tahun 2008, telah menjadi sistem operasi paling populer untuk perangkat seluler pintar (Kaijun Liu, 2022). Sehingga aplikasi ini dibuat menggunakan software *Android Studio* dengan versi *Android SDK Build-Tool*s versi 30.0.0rc4 dan *Android SDK Platform-Tools* versi 30.0.1. Spesifikasi minimal perangkat *android* untuk menjalankan aplikasi ini adalah *SDK* 21 atau *android* versi 5.0 (*Lollipop*), mempunyai kamera belakang yang berfungsi serta *flashlight*. Aplikasi ini tidak membutuhkan koneksi internet, jadi dapat dijalankan secara *offline*.

SATIN – Sains dan Teknologi Informasi
Vol. 8, No. 1, Juni 2022, pp. 12-22
ISSN: 2527-9114, DOI: 10.33372/stn.v8i1.816



Gambar 14. Tampilan aplikasi

Pada saat aplikasi dijalankan, maka *flashlight* akan otomatis menyala dan aplikasi mempunyai tampilan utama yang terdiri atas 2 bagian, yaitu kamera dan informasi. Bagian kamera merupakan background dari tampilan diatas yang berwarna hitam, sedangkan bagian informasi merupakan bagian yang didalamnya terdapat text nominal uang dan persentase keyakinannya, dimana akan berubah pada saat aplikasi dijalankan. Misalnya nominal uang berganti menjadi "seribu" dan persentase keakuratannya berganti menjadi "100%". *Interference Time* adalah penunjuk berapa lama waktu yang dibutuhkan untuk mengenali objek (nilai mata uang), pada saat aplikasi dijalankan akan muncul hasilnya dalam format ms (*millisecond*), misalnya 20ms. Khusus pada bagian informasi nominal dan persentase-nya hanya akan muncul apabila nilai persentase keakuratannya minimal 100%. Apabila keakuratannya dibawah 100%, maka informasi nominal dan persentase tidak akan muncul atau *blank*. Pada saat nominal uang dikenali dengan keakuratan minimal 100%, maka aplikasi akan otomatis memberikan suara atau *output* berupa jumlah nominal uang tersebut.

E. Pengujian dan Evaluasi Aplikasi

Pengujian aplikasi dilakukan uang terhadap beberapa gambar uang kertas rupiah yang terdiri dari seribu, duaribu, limaribu, sepuluhribu, duapuluhribu, limapuluhribu dan seratusribu dan dalam berbagai kondisi dan pencahayaan dengan terang (dengan *flash*) dan tanpa *flash*. Setelah di uji sebanyak 20 kali untuk masing-masing nominal mata uang dan masing-masing kondisi cahaya, kemudian dilakukan evaluasi dengan *Confusion Matrix* berdasarkan hasil dari penguijian aplikasi. *Confusion Matrix* melakukan pengujian untuk memprediksi suatu objek yang benar dan salah (Rozzi Kesuma Dinata, Safwandi, Novia Hasdyna, 2020).

Tabel 1. Evaluasi Confusion Matrix dengan Flash

	Actual Values						
	1000	2000	5000	10000	20000	50000	100000
1000	19	0	0	0	0	0	0
2000	1	19	0	0	0	0	0
5000	0	1	20	0	0	0	0
10000	0	0	0	20	0	0	0
20000	0	0	0	0	20	0	0
50000	0	0	0	0	0	20	0
100000	0	0	0	0	0	0	20
Total	**20**	**20**	**20**	**20**	**20**	**20**	**20**
Akurasi	**0,95**	**0,95**	**1**	**1**	**0,95**	**1**	**1**
Precision	**1**	**0,95**	**0,95**	**1**	**1**	**0,95**	**1**	**1**
Recall	**0,95**	**0,59**	**1**	**1**	**0,95**	**1**	**1**

Total *Accuracy* = $\frac{\sum Accuracy}{n\,Accuracy} = \frac{(0{,}95 + 0{,}95 + 1 + 1 + 0{,}95 + 1 + 1)}{7} = \frac{6{,}85}{7} = 0{,}98$ (1)

Total *Precision* = $\frac{\sum Precision}{n\,Precision} = \frac{(1 + 0{,}95 + 0{,}95 + 1 + 1 + 0{,}95 + 1)}{7} = \frac{6{,}85}{7} = 0{,}98$ (2)

SATIN – Sains dan Teknologi Informasi
Vol. 8, No. 1, Juni 2022, pp. 12-22
ISSSN: 2527-9114, DOI: 10.33372/stn.v8i1.816

Total *Recall* = $\frac{\sum Recall}{n\,Recall} = \frac{(0{,}95 + 0{,}95 + 1 + 1 + 0{,}95 + 1 + 1)}{7} = \frac{6{,}85}{7} = 0{,}98$ (3)

Tabel 2. Evaluasi Confusion Matrix dengan Flash

	Actual Values						
	1000	2000	5000	10000	20000	50000	100000
1000	19	0	0	0	0	0	0
2000	1	17	0	0	0	0	3
5000	0	3	19	2	0	0	3
10000	0	0	0	14	0	0	0
20000	0	0	1	2	19	0	5
50000	0	0	0	2	1	20	9
100000	0	0	0	0	0	0	0
Total	**20**	**20**	**20**	**20**	**20**	**20**	**20**
Akurasi	**0,95**	**0,85**	**0,95**	**0,7**	**0,95**	**1**	**0**
Precision	**1**	**0,85**	**0,68**	**0,7**	**0,7**	**0,62**	**0**
Recall	**0,95**	**0,85**	**0,95**	**0,7**	**0,95**	**1**	**0**

Total *Accuracy* = $\frac{\sum Accuracy}{n\,Accuracy} = \frac{(0{,}95 + 0{,}85 + 0{,}95 + 0{,}7 + 0{,}95 + 1 + 0)}{7} = \frac{5{,}4}{7} = 0{,}77$ (4)

Total *Precision* = $\frac{\sum Precision}{n\,Precision} = \frac{(1 + 0{,}85 + 0{,}68 + 0{,}7 + 0{,}7 + 0{,}62 + 0)}{7} = \frac{4{,}55}{7} = 0{,}65$ (5)

Total *Recall* = $\frac{\sum Recall}{n\,Recall} = \frac{(0{,}95 + 0{,}85 + 0{,}95 + 0{,}7 + 0{,}95 + 1 + 0)}{7} = \frac{5{,}4}{7} = 0{,}77$ (6)

Nilai *accuracy* adalah nilai atau tingkat kedekatan antara nilai yang diprediksi terhadap nilai sesunggunya. *Precision* adalah tingkat ketepatan antara informasi yang diminta dengan jawaban yang diberikan oleh aplikasi. *Recall* adalah tingkat keberhasilan sistem menemukan kembali sebuah informasi. Dimana pada penelitian ini, aplikasi dengan *flash* menyala menghasilkan baik *accuracy, precision* dan *recall* sebesar 0,98. Sedangkan aplikasi dengan *flash* tidak menyala mendapatkan *accuracy* sebesar 0.77, *precision* sebesar 0.65 dan *recall* sebesar 0.77.

4. Simpulan

Metode *YOLOv3 (You Only Look Once*) cocok digunakan untuk melatih *data set* yang telah dibuat menjadi sebuah model yang siap untuk diimplementasikan ke bentuk aplikasi seperti *android* yang dapat berjalan serta berfungsi dengan baik dengan memberikan *output* berupa suara agar penyandang tunanetra dapat mengetahui hasilnya. Masalah yang dapat mengurangi keakuratan hasil adalah pencahayaan pada saat objek diambil, di mana hasilnya perbandingannya adalah dengan *flash* menyala menghasilkan *accuracy, precision* dan *recall* sebesar 0,98. Sedangkan aplikasi dengan *flash* tidak menyala menghasilkan *accuracy* dan *recall* sebesar 0,77 dan *precision* sebesar 0,65.

5. Referensi

Abadi, M., Barham, P., Chen, J., & Chen, Z. (2016). Tensorflow: A System For Large-Scale Machine Learning Mart´In. *Proceedings Of The 12th Usenix Symposium On Operating Systems Design And Implementation*. Https://Doi.Org/10.1016/0076-6879(83)01039-3

Adie; Renata Tresy, H. (2018). *Pengenalan Objek Pada Citra Digital Dengan Algoritma Region-Based Convolutional Neural Network (R-Cnn) - E-Journal Universitas Atma Jaya Yogyakarta*.

Andre Kanisius Edguard Lapian, S. R. U. . S. D. K. M. (2021). Implementasi Framework You Only Look Once

Untuk Klasifikasi Pola Tanda Tangan | Lapian | Jurnal Teknik Informatika. *Jurnal Teknik Informatika*, *16*(3), 337–346. Https://Ejournal.Unsrat.Ac.Id/Index.Php/Informatika/Article/View/34223/33692

Anjar Wanto. (2018). Penerapan Jaringan Saraf Tiruan Dalam Memprediksi Jumlah Kemiskinan Pada Kabupaten/Kota Di Provinsi Riau | Wanto | Klik - Kumpulan Jurnal Ilmu Komputer. *Jurnal Ilmiah Klik Kumpulan Jurnal Ilmu Komputer*, *5*(1), 61–74. Http://Klik.Ulm.Ac.Id/Index.Php/Klik/Article/View/129/Pdf

Cross, S. S., Harrison, R. F., & Kennedy, R. L. (1995). An Introduction To Neural Networks. In *The Lancet* (Vol. 346, Issue 8982). Https://Doi.Org/10.1016/S0140-6736(95)91746-2

Dey, A. (2016). Machine Learning Algorithms: A Review. *International Journal Of Computer Science And Information Technologies*, *7*(3), 1174–1179.

Hadiwandra, T. Y. (2019). Perbandingan Kinerja Model Klasifikasi Decission Tree , Bayesian Classifier, Instance Base, Linear Function Base, Rule Base Pada 4 Dataset Berbeda. *Satin - Sains Dan Teknologi Informasi*, *5*(1), 70–78. Https://Doi.Org/10.33372/Stn.V5i1.452

Informasi Kementrian Kesehatan Ri, P. D. (2019). Situasi Disabilitas. *Pusat Data Dan Informasi Kementrian Kesehatan Ri*, 1–10.

Kaijun Liu, S. X. X. Z. S. A. H. L. (2022). A Review Of Android Malware Detection Approaches Based On Machine Learning. *Ieeeaccess*, *7*, 124579–124607. Https://Ieeexplore.Ieee.Org/Stamp/Stamp.Jsp?Tp=&Arnumber=9130686

Ma, H., Liu, Y., Ren, Y., & Yu, J. (2020). Detection Of Collapsed Buildings In Post-Earthquake Remote Sensing Images Based On The Improved Yolov3. *Remote Sensing*, *12*(1). Https://Doi.Org/10.3390/Rs12010044

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep Learning Applications And Challenges In Big Data Analytics ." Journal Of Big Data. *Journal Of Big Data*, *2*(1), 1–21. Https://Doi.Org/10.1186/S40537-014-0007-7

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. *In Proceedingsof The Ieee Conference On Computer Vision And Pattern Recognition(Cvpr)*. Https://Doi.Org/10.1021/Je00029a022

Rozzi Kesuma Dinata, Safwandi, Novia Hasdyna, N. A. (2020). View Of Analisis K-Means Clustering Pada Data Sepeda Motor. *Informatics Journal*, *5*(1), 10–17. Https://Jurnal.Unej.Ac.Id/Index.Php/Informal/Article/View/17071/8199

Shinde, S., Kothari, A., & Gupta, V. (2018). Yolo Based Human Action Recognition And Localization. *Procedia Computer Science*, *133*(2018), 831–838. Https://Doi.Org/10.1016/J.Procs.2018.07.112

Sianturi, J., & Hajjah, A. (2021). Analisis Sentimen Prosesor Amd Ryzen Menggunakan Metode Support Vector Machine. *Satin-Sains Dan Teknologi Informasi*, *7*(2), 129–141. Https://Doi.Org/10.33372/Stn.V7i2.804

Simeone, O. (2018). A Brief Introduction To Machine Learning For Engineers. *Foundations And Trends In Signal Processing*, *12*(3–4), 200–431. Https://Doi.Org/10.1561/2000000102

Yuni Eka Achyani. (2018). Penerapan Metode Particle Swarm Optimization Pada Optimasi Prediksi Pemasaran

Langsung | Achyani | Jurnal Informatika. *Jurnal Informatika*, *5*(1), 1–11. Https://Ejournal.Bsi.Ac.Id/Ejurnal/Index.Php/Ji/Article/View/2736/Pdf