



IMPLEMENTASI DEEP LEARNING DENGAN METODE CONVOLUTIONAL NEURAL NETWORK UNTUK IDENTIFIKASI OBJEK SECARA REAL TIME BERBASIS ANDROID

Indra Fransiskus Alam^{*1}, Muhammad Ihsan Sarita², Adha Mashur Sajiah³

^{1,2,3}Jurusan Teknik Informatika, Fakultas Teknik, Universitas Halu Oleo, Kendari

e-mail: ^{*1}indrafransiskusalam@gmail.com, ²ihsansarita@yahoo.co.id, ³adha.m.sajiah@uho.ac.id

Abstrak

Perkembangan kecerdasan buatan atau *Artificial Intelligence* saat ini telah mengalami perubahan yang signifikan. Hal tersebut mendasari lahirnya metode untuk mengatasi deteksi objek secara real time dengan akurasi yang tinggi. Pada dasarnya *Deep Learning* adalah implementasi konsep dasar dari *Machine Learning* yang menerapkan algoritma pintar dengan lapisan yang lebih banyak antara lapisan masukan dan lapisan keluaran. *Convolutional Neural Network* (CNN) merupakan penerapan untuk sebuah *Deep learning* (DL) yang dapat digunakan untuk mendeteksi dan mengenali sebuah objek pada sebuah citra digital. Kemampuan CNN diklaim sebagai model terbaik untuk memecahkan permasalahan *object detection* dan *object recognition* karena merupakan pengembangan dari metode *backpropagation* dan tidak memerlukan komputasi yang besar dalam prosesnya. Hasil yang didapatkan pada penelitian ini bahwa aplikasi *android* dapat berjalan baik dengan akurasi sebesar 92,33% dapat dilihat dari hasil pengujian dengan menggunakan metode *10-fold cross validation*, semua menu yang ada dapat berjalan dan penyebutan label objek sesuai untuk pengenalan dan klasifikasi citra. Perhitungan *precision* dan *recall* memiliki nilai yang baik, masing-masing sebesar 97,51% dan 94,33%.

Kata kunci — *Deep Learning, Convolutional Neural Network, AutoML, Text-to-speech*

Abstract

The development of artificial intelligence has now undergone significant changes. This underlies the birth of a method to deal with object detection in real time with high accuracy. Basically Deep Learning is the implementation of the basic concept of Machine Learning which applies the smart algorithm with more layers between the input layer and the output layer. Convolutional Neural Network (CNN) is one method of Deep learning (DL) that can be used to detect and recognize an object in a digital image. The ability of CNN is claimed to be the best model to solve the problem of object detection and object recognition because it is a development of the backpropagation method and does not require large computation in the process. The results obtained in this study that android applications can run well with an accuracy of 92.33% can be seen from the test results using the 10-fold cross validation method, all available menus can be run and the mention of object labels is appropriate for image recognition and classification. Calculation of precision and recall has good values, each at 97.51% and 94.33%.

Keywords—*Deep Learning, Convolutional Neural Network, AutoML, Text-to-speech*



1. PENDAHULUAN

Seiring dengan kemajuan zaman, klasifikasi citra digital sangat dibutuhkan di berbagai macam bidang, seperti: informatika, kedokteran, kelautan, pertanian, dan bisnis. Klasifikasi citra saat ini menjadi salah satu problem yang telah lama dicari solusinya dalam *computer vision*. Bagaimana menduplikasikan kemampuan manusia dalam memahami informasi citra digital, supaya komputer dapat mengenali objek pada citra selayaknya manusia. Proses *feature engineering* yang digunakan pada umumnya sangat terbatas dimana hanya dapat berlaku pada dataset tertentu saja tanpa kemampuan generalisasi apapun.

Rumus empiris terbaik dari hasil permodelan *Artificial Neural Network* yang dapat digunakan memperkirakan biaya pembangunan gedung kampus pada tahap konseptual untuk prakiraan biaya dengan konfigurasi ANN 6-9-1 (enam variable *input*), satu *hidden layer* dengan sembilan *neuron* dan satu *output*). Prosentase *error* / MMRE maksimal yang dihasilkan adalah sebesar 6.39%. Sementara hasil penelitian Kesturi (2012) Estimasi Biaya Tahap Konseptual pada Konstruksi Gedung Perkantoran dengan Metode ANN, meneliti terkait penggunaan model arsitektur ANN untuk estimasi biaya pada gedung perkantoran memiliki nilai deviasi MMRE lebih baik sebesar 7.79% dari kriteria deviasi tahap konseptual menurut AACE (*American Association of Cost Engineering*), yaitu, kelas 4: dengan *range* deviasi -15% - -30% dan +25%+50% [1].

Deep Learning adalah sub bidang dari *Machine Learning*. Pada dasarnya *Deep Learning* adalah implementasi konsep dasar dari *Machine Learning* yang menerapkan algoritma ANN dengan lapisan yang lebih banyak. Banyaknya lapisan tersembunyi yang digunakan antara lapisan masukan dan lapisan keluaran, maka jaringan ini dapat dikatakan *deep neural net*. Hal ini sebagian besar dipengaruhi faktor komputasi quantum, data set yang besar dan teknik untuk melatih jaringan yang lebih dalam.

Pada daftar beberapa *framework deep learning* yang tersedia saat ini, *framework* bekerja berdasarkan beberapa fitur, seperti pengamatan dari sudut pandang, rancangan antar muka, serta dukungan penuh dari

beberapa metode, seperti *Convolutional Neural Network* (CNN), *Recurrent Neural Network* (RNN), *Restricted Boltzmann Machine* (RBM) [2].

Convolutional Neural Network (CNN) adalah salah satu metode penerapan *Deep learning* yang dapat digunakan untuk mendeteksi dan mengenali sebuah objek pada sebuah citra digital. Kemampuan CNN di klaim sebagai model terbaik untuk memecahkan permasalahan *object detection* dan *object recognition* karena merupakan pengembangan dari metode *backpropagation* dan tidak memerlukan komputasi yang besar dalam prosesnya. Sedangkan metode RNN digunakan untuk memproses dengan suatu sampel dalam urutan mempunyai hubungan erat satu dengan yang lain (Sinyal audio, penerjemah bahasa, dan lain-lain) dan RBM digunakan untuk pengambilan keputusan dan prediksi [3].

Sehubungan dengan metode tersebut, tim Google merancang sebuah kecerdasan buatan yang dinamai dengan *AutoML* yang mampu membuat kecerdasan buatan sendiri untuk perusahaan. *AutoML* dirancang menggunakan strategi pembelajaran yang dilakukan secara mendalam dan dengan data transmisi jaringan saraf yang luas. Para peneliti telah mampu membuat algoritma belajar yang dapat mengotomasi salah satu bagian paling sulit dari pekerjaan merancang perangkat lunak mesin belajar (*Machine Learning*) untuk menangani tugas tertentu.

Berdasarkan pemaparan pada paragraf sebelumnya dan melihat permasalahan yang ada maka penulis akan mengimplementasikan dari metode *deep learning* menggunakan metode *Convolutional Neural Network* (CNN) dan *Google AutoML* untuk membantu mengenali dan mengidentifikasi objek dengan menggunakan teknologi yang berbasis Android.

2. METODE PENELITIAN

2.1 Deep Learning

Deep Learning adalah bidang yang muncul dari penelitian pembelajaran mesin (ML) yang terdiri dari beberapa lapisan tersembunyi dari jaringan saraf tiruan. Metodologi *Deep Learning* menerapkan transformasi *nonlinier* dan model abstraksi tingkat tinggi dalam basis data besar.

Kemajuan terbaru dalam arsitek pembelajaran dalam di banyak bidang telah memberikan kontribusi signifikan dalam kecerdasan buatan. Konsep *Deep Learning (DL)* muncul untuk pertama kalinya pada tahun 2006 sebagai bidang baru dari penelitian dalam pembelajaran mesin.

Pertama kali dikenal sebagai pembelajaran hierarkis, dan biasanya melibatkan banyak bidang penelitian yang berkaitan dengan pengenalan pola. Pembelajaran mendalam terutama mempertimbangkan dua faktor utama: pemrosesan *nonlinear* dalam berbagai lapisan atau tahapan dan pembelajaran yang diawasi (*supervised*) atau tidak diawasi (*unsupervised*). Pemrosesan *nonlinear* dalam berbagai lapisan merujuk ke suatu algoritma yang mengambil *output* dari lapisan sebelumnya sebagai sebuah *input* [4].

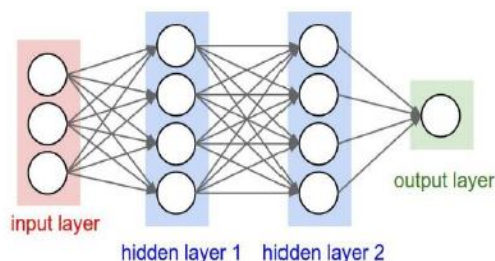
2.2 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah pengembangan dari *Multilayer Perceptron (MLP)* yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. Pada kasus klasifikasi citra, MLP kurang sesuai untuk digunakan karena tidak menyimpan informasi spasial dari data citra dan menganggap setiap piksel adalah fitur yang independen sehingga menghasilkan hasil yang kurang baik.

CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kunihiko Fukushima, seorang peneliti dari NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari AT&T Bell Laboratories di Holmdel, New Jersey, USA. Model CNN dengan nama LeNet berhasil diterapkan oleh LeChun pada penelitiannya mengenai pengenalan angka dan tulisan tangan.

Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi ImageNet Large Scale Visual Recognition Challenge 2012. Prestasi tersebut menjadi momen pembuktian bahwa metode *Deep Learning*, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode *Machine Learning* lainnya seperti SVM pada kasus klasifikasi objek pada citra.

Cara kerja CNN memiliki kesamaan pada MLP, namun dalam CNN setiap neuron dipresentasikan dalam bentuk dua dimensi, tidak seperti MLP yang setiap neuron hanya berukuran satu dimensi. Arsitektur MLP dijelaskan pada Gambar 1.



Gambar 1 Arsitektur MLP Sederhana

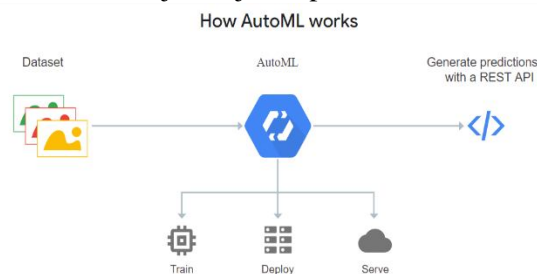
Sebuah MLP memiliki i *layer* (kotak merah dan biru) dengan masing-masing *layer* berisi j i neuron (lingkaran putih). MLP menerima *input* data satu dimensi dan mempropagasikan data tersebut pada jaringan hingga menghasilkan *output*. Setiap hubungan antar *neuron* pada dua *layer* yang bersebelahan memiliki parameter bobot satu dimensi yang menentukan kualitas mode. Disetiap data input pada layer dilakukan operasi *linear* dengan nilai bobot yang ada, kemudian hasil komputasi akan ditransformasi menggunakan operasi *non linear* yang disebut sebagai fungsi aktivasi [5].

2.3 Google AutoML

API Google *AutoML* memungkinkan pengembang untuk memahami konten gambar dengan mengenkapsulasi model pembelajaran mesin yang canggih dengan API REST yang mudah digunakan. Dengan cepat mengklasifikasikan gambar ke dalam ribuan kategori (mis., "Perahu layar", "singa", "Menara Eiffel"), mendeteksi objek dan wajah individual dalam gambar, dan menemukan dan membaca kata-kata tercetak yang terdapat di dalam gambar.

AutoML dapat membangun metadata pada katalog gambar, konten ofensif yang moderat, atau mengaktifkan skenario pemasaran baru melalui analisis sentimen gambar. Analisis gambar yang diunggah dalam permintaan atau integrasikan dengan penyimpanan gambar di *Google Cloud Storage*. *Cloud AutoML* adalah rangkaian produk pembelajaran mesin yang memungkinkan pengembang dengan keahlian

pembelajaran mesin terbatas untuk melatih model berkualitas tinggi khusus untuk kebutuhan bisnis mereka. Ini bergantung pada pembelajaran transfer Google yang mutakhir dan teknologi pencarian arsitektur neural. *AutoML* bekerja ditunjukkan pada Gambar 2.



Gambar 2 Alur kerja *AutoML*.

AutoML merupakan *machine learning products* yang dapat memungkinkan para pengembang aplikasi untuk membantu dalam melatih model yang memiliki kualitas yang tinggi, agar dapat digunakan sesuai dengan kebutuhan pasar. *Cloud AutoML* sudah dikembangkan selama lebih dari 10 tahun untuk menghasilkan performa yang sangat cepat dan prediksi yang lebih akurat. Tampilan yang terdapat pada *AutoML* terdiri dari *train*, *evaluate*, *improve*, dan *deploy* model sesuai dengan data yang ada. Serta terdapat fitur *Google's human labelling* yang dapat membantu menentukan label dari citra atau data secara otomatis [6].

2.4 Text-to-speech

Secara umum pengertian *Text-To-Speech* adalah sebuah sistem yang mengubah suatu teks menjadi bentuk ucapan. Menurut beberapa literatur, pengertian TTS adalah sebagai produksi ucapan secara otomatis melalui transkripsi *grapheme to phoneme* dari sebuah kalimat. Pada prinsipnya, TTS terdiri atas dua sub sistem, yaitu *converter* teks ke fonem (*text to phoneme*) dan *converter* fonem ke ucapan (*phoneme to speech*). Bagian *converter* teks ke fonem berfungsi untuk mengubah kalimat masukan dalam suatu bahasa tertentu yang berbentuk teks menjadi rangkaian kode-kode bunyi yang biasanya direpresentasikan dengan kode fonem, durasi, serta *pitch*-nya. Bagian ini bersifat sangat *language dependant*. Untuk suatu bahasa baru, bagian ini harus dikembangkan secara lengkap khusus untuk bahasa tersebut. Bagian *converter* fonem ke ucapan menerima masukan berupa kode-kode fonem serta *pitch*

dan durasi yang dihasilkan oleh bagian sebelumnya. Berdasarkan kode-kode tersebut, bagian *converter* fonem ke ucapan menghasilkan bunyi atau sinyal ucapan yang sesuai dengan kalimat yang ingin diucapkan [7].

2.5 Joint Photographic Experts Group (JPEG)

Dalam ilmu komputer, JPEG (baca: /'dʒeɪpɛɡ/) adalah metode kompresi yang umum digunakan untuk gambar-gambar fotografi. JPEG merupakan singkatan dari *Joint Photographic Experts Group*, nama dari komite yang menetapkan standar JPEG. Pada tahun 1994, standar JPEG disahkan sebagai ISO 10918-1. Standar JPEG memberikan spesifikasi *codec* kompresi data ke dalam *stream* data *byte* dan pendekompresian kembali ke bentuk gambar serta format data penyimpanannya. Metode kompresi data yang digunakan umumnya berupa *lossy compression*, yang membuang detail visual tertentu saat dilakukan proses kompresi, dimana dengan pembuangan detail visual tertentu tersebut akan menyebabkan hilangnya data tersebut tidak bisa dikembalikan dan akan berakibat menurunkan kualitas citra.

Namun, karena JPEG mendukung indeks warna sebanyak 16 juta sehingga penurunan kualitas terhadap citra JPEG tidak akan terlihat secara jelas oleh mata manusia. File JPEG memiliki ekstensi .jpg, .jpeg, .jpe, .jfif, dan .jif. Format JPEG banyak digunakan untuk penyimpanan dan transfer data fotografi lewat internet. Dalam hal ini format JPEG jauh lebih baik dari GIF yang menggunakan *pallette* maksimum 256 warna [8].

3. HASIL DAN PEMBAHASAN

3.1 Implementasi

Implementasi sistem merupakan sebuah perancangan sistem yang dibuat dalam bentuk nyata. Sistem ini dibuat dengan bahasa pemrograman Kotlin, dengan menggunakan Android Studio IDE. Aplikasi pesan *chatting* dirancang dengan mengimplementasikan *Google Cloud AutoML*.

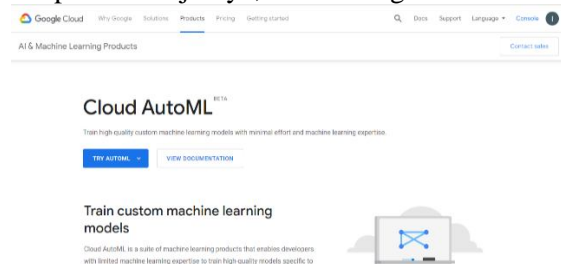
1. Interface Aplikasi

Pada *interface* Aplikasi di antaranya :

a. Halaman Utama

Pada tampilan ini adalah halaman utama pada saat *admin* membuka *Google Cloud AutoML*. Pengguna menekan tombol *Try*

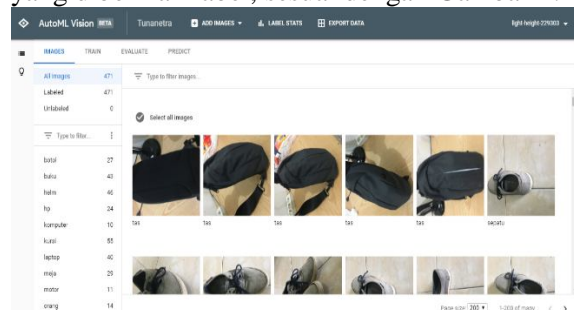
AutoML Vision, dan login menggunakan akun *gmail admin*. Setelah itu akan masuk pada tampilan selanjutnya, sesuai dengan Gambar 3.



Gambar 3 Alur kerja *AutoML*.

b. Halaman *Images*

Halaman ini berisi kumpulan citra (botol, buku, tas, kursi, laptop, dan lain-lain) yang diberikan label, sesuai dengan Gambar 4.



Gambar 4 Alur kerja *AutoML*.

c. Halaman *Train*

Pada tampilan ini menampilkan halaman *train* yang akan digunakan oleh *admin* untuk melatih *dataset* dalam sebuah model dengan menekan tombol *Train New Model*, dan melihat rincian *dataset* yang pernah dijadikan model, sesuai dengan Gambar 5.



Gambar 5 Halaman *Train*

d. Halaman *Evaluate*

Pada tampilan ini menampilkan hasil dari evaluasi *training model*. Terdapat rincian total *images* yang berisi jumlah seluruh citra *training*, *precision* yang berisi pengetahuan dari semua contoh uji yang diberikan label, berapa banyak sebenarnya yang seharusnya dikategorikan dengan label yang ada, *recall* memberikan informasi mengenai persentase tingginya *false negative* dari *model*, *score*

threshold berfungsi sebagai perbandingan hasil *training*.

Jika *score threshold* rendah, maka *model* akan mengklasifikasikan lebih banyak gambar, tetapi berisiko menghasilkan klasifikasi yang kurang *valid*. Jika *score threshold* tinggi, *model* akan mengklasifikasikan lebih sedikit gambar, namun dapat menghasilkan klasifikasi yang cukup *valid* yang ditujukan pada Gambar 6. Sedangkan *Confusion matrix* memperlihatkan hasil perbandingan *true positive* dan *true negative*, sesuai dengan Gambar 7.



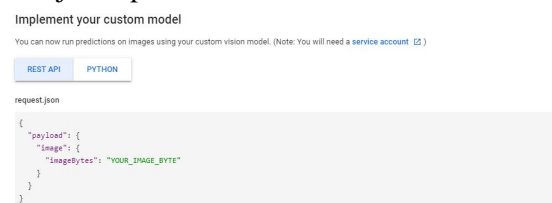
Gambar 6 Halaman *Evaluate*

True label \ Predicted \ label	motor	botol	tas	komputer	laptop	sendok	pulpen	orang	kursi	sepatu
motor	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
botol	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
tas	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
komputer	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
laptop	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%	0.0%
sendok	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%
pulpen	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%	0.0%
orang	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%	0.0%
kursi	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%	0.0%
sepatu	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	100.0%

Gambar 7 *Confusion matrix*

e. Halaman *Predict*

Pada tampilan ini berisi format penulisan *API* yang berbentuk *.json* dan *.py*. Berfungsi sebagai komunikasi data kepada perangkat atau *device* yang lain, sesuai yang ditunjukkan pada Gambar 8.

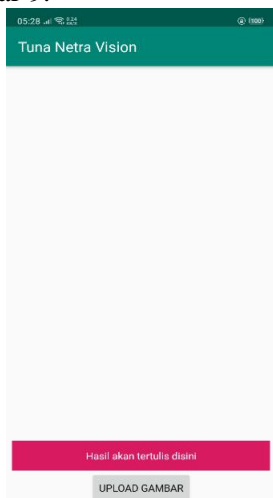


Gambar 8 Halaman *Predict*

f. *Menu Android*

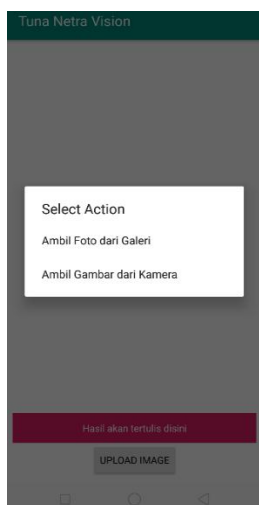
Pada tampilan ini menampilkan tombol “Upload Gambar” untuk memasukkan citra

objek yang diinginkan, sesuai yang ditunjukkan pada Gambar 9.



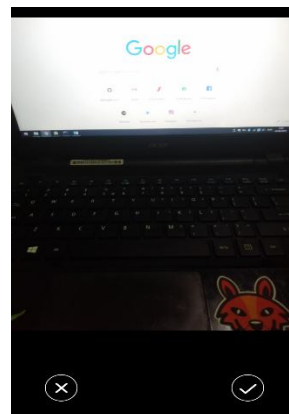
Gambar 9 Tampilan Menu Android

- g. Tampilan Pilihan Sumber Citra
 Pada tampilan ini, *user* dapat memilih metode penggunaan *upload* citra, antara lain memilih dari penyimpanan *android* atau langsung mengambil citra secara realtime dari kamera *handphone*, sesuai yang ditunjukkan pada Gambar 10.



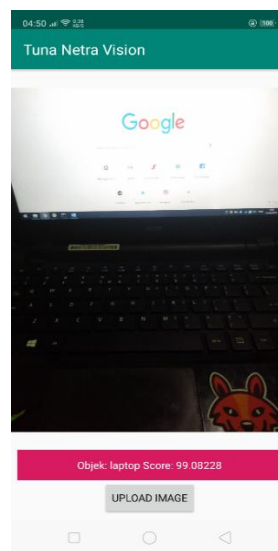
Gambar 10 Tampilan Menu Sumber Citra

- h. Tampilan Kamera
 Pada tampilan ini, *user* dapat memilih lokasi, pencahayaan, dan objek yang diinginkan, sesuai yang ditunjukkan pada Gambar 11. Setelah itu menggunakan fitur ambil gambar dan klik tanda ceklis pada layar android.



Gambar 11 Tampilan Kamera

- i. Tampilan Hasil
 Pada tampilan ini, *user* dapat mengetahui objek apa yang ditangkap kamera dengan bantuan suara berbahasa Indonesia dan label objek tersebut, sesuai yang ditunjukkan pada Gambar 12.



Gambar 12 Tampilan Hasil

3.2 Pengujian Sistem

Pada hasil rekap sepuluh pengujian, setelah memberikan sampel citra acak sebanyak 5 citra untuk data *training* dan 3 untuk data testing pada *dataset*. Diketahui bahwa semua objek mendapatkan persentase keakuratan sebesar 92,33%, perhitungan *precision* didapatkan hasil sebesar 97,51%, *recall* sebesar 94,33% untuk pengujian *cross validation* tahap.

Tabel 1. Hasil Rekap Pengujian

Pengujian	<i>Precision</i>	<i>Recall</i>	<i>Accuration</i>
1	96,55%	93,33%	90,32%
2	92,59%	83,33%	78,12%
3	96,67%	96,67%	93,54%
4	96,43%	90,00%	87,10%
5	96,43%	90,00%	87,10%
6	100%	100%	87,10%
7	100%	100%	100%
8	100%	100%	100%
9	100%	100%	100%
10	100%	100%	100%
Rata-rata	97,51%	94,33%	92,33%

4. KESIMPULAN

Berdasarkan penelitian dan hasil pengujian yang dilakukan pada metode *Convolutional Neural Network* dan *Google Cloud AutoML*, maka disimpulkan sebagai berikut:

1. Algoritma CNN (*Convolutional Neural Network*) dan *AutoML* dapat diimplementasikan pada aplikasi atau sistem untuk mengenali objek dengan nilai *precision* sebesar 97,51% dan *recall* sebesar 94,33%, untuk pengenalan objek pada masing-masing benda.
2. Akurasi rata-rata semua benda dengan menggunakan metode *10-fold cross validation* sebesar 92,33%.
3. Proses pengambilan citra dan hasil yang dikeluarkan tergolong mudah untuk dipahami karena dapat digunakan pada *smartphone*, serta membantu dalam identifikasi objek yang tunggal maupun banyak.

5. SARAN

Saran yang dapat diberikan dalam pengembangan selanjutnya adalah sebagai berikut:

1. Menggunakan metode pembaruan dari CNN (*Convolutional Neural Network*) untuk proses yang lebih cepat dengan skala yang lebih luas.
2. Sebaiknya menggunakan bantuan *library* seperti *tensorflow*, *ML kit*, dan lain-lain untuk memudahkan dalam *training model* dan *testing* secara *offline*.

DAFTAR PUSTAKA

- [1] A. Y. Prathama, A. Aminullah, and A. Saputra, "Pendekatan ANN (Artificial Neural Network) untuk Penentuan Prosentase Bobot Pekerjaan Pada Rumah Sakit Pratama," *J. Tekno Sains*, Vol. 7, No. 1, pp. 14–25, 2017.
- [2] A. R. Pathak, M. Pandey, and S. Rautaray, "Application of Deep Learning for Object Detection," *Procedia Comput. Sci.*, Vol. 132, pp. 1706–1717, 2018.
- [3] I. Goodfellow, Y. Bengio, and A. Courville, *Adaptive Computation and Machine Learning*. London, England: MIT Press, 2016.
- [4] A. Mosavi, R. Vargas, and L. Ruiz, "Deep Learning : A Review," *Advances in Intelligent Systems and Computing*, 2017. [Online]. Available: https://www.researchgate.net/publication/n/318447392_DEEP_LEARNING_A_REVIEW. [Accessed: 05-Jul-2019].
- [5] A. Y. Wicaksono, N. Suciati, C. Fatichah, K. Uchimura, and G. Koutaki, "Implementasi Deep Learning Untuk Image Classification Menggunakan Algoritma Convolutional Neural Network (Cnn) Pada Citra Wayang Golek," Universitas Islam Indonesia, 2018.
- [6] A. A. Borkowski *et al.*, "Google Auto ML versus Apple Create ML for Histopathologic Cancer Diagnosis ; Which Algorithms Are Better?," 2018. [Online]. Available: <https://arxiv.org/abs/1903.08057>. [Accessed: 01-Jul-2019].
- [7] R. A. W, H. Tolle, and O. Setyawati, "Pengembangan Aplikasi Text - to - Speech Bahasa Indonesia Menggunakan Metode Finite State Automata Berbasis Android," *JNTETI*, Vol. 5, No. 1, pp. 1–8, 2016.
- [8] J. Ding, S. Pei, W. Wei, H. Chen, and T. Lee, "Adaptive Golomb Code for

Joint Geometrically Dis-tributed Data and Its Application in Image Coding,” in *Proceedings of the Second APSIPA Annual Summit and Conference*, 2010, pp. 302–305.
