



Klasifikasi Jenis Tas Pada Gambar 360 Derajat (*Fish Eye*) Dengan Menggunakan *Tensorflow*

(*Classification Of Bag Types In 360 Degree Images (Fish Eye) Using Tensorflow*)

Agnes Novi Anna Pangemanan¹, Arief Suryadi Satyawan², Sri Desy Siswanti³

¹ Universitas Nurtanio

E-mail: pangemanan3610@gmail.com

² Badan Riset dan Inovasi Nasional

E-mail: arief.suryadi@akane.waseda.jp

³ Universitas Nurtanio

desy0712unnur@gmail.com

Abstract— *Classification of objects is one of the studies that are currently being developed. The impact on the world of fashion is where women and men, teenagers to parents today cannot be separated from the bag as an addition to daily fashion. In the selection of bags, mistakes are often made, so as not to make a wrong choice, in this final project, I classify the types of bags, which is the method used to distinguish the characteristics of bags by type. The bag classification system based on type is a program that can identify a person's bag according to the type that has been trained and stored in the database of the program being run. Classification of bag types can be done in various ways, one of which is Deep Learning with the Convolutional Neural Network (CNN) method, CNN implementation using Tensorflow with the python programming language. This study was conducted using 5 classifications of bag type datasets totaling 6,720 images that have been trained with an image size of 180 x 180 using a 360° camera. It is hoped that this system is able to work well for classifying bag types in 360° (fish eye) image format. This study resulted in true detection rates of 55% and false detection of 45% where true detection is seen from the number of truths of accuracy in determining the output results, while false detection is the opposite of true detection from the number of 135 images that have been tested.*

Keywords— *Deep Learning, Convolutional Neural Network, Tensorflow, Camera 360°, Bag Types, Classification.*

Abstrak— *Klasifikasi objek merupakan salah satu kajian yang sedang berkembang saat ini. Dampaknya pada dunia fashion dimana wanita dan pria, remaja hingga orang tua saat ini tidak lepas dari tas sebagai pelengkap fashion sehari-hari. Dalam pemilihan tas sering terjadi kesalahan agar tidak salah dalam memilih, pada tugas akhir ini saya mengklasifikasikan jenis-jenis tas, yaitu cara yang digunakan untuk membedakan karakteristik tas berdasarkan jenisnya. Sistem klasifikasi tas berdasarkan jenisnya merupakan program yang dapat mengidentifikasi tas seseorang sesuai dengan jenisnya yang telah dilatih dan disimpan dalam database program yang sedang dijalankan. Klasifikasi jenis tas dapat dilakukan dengan berbagai cara salah satunya Deep Learning dengan metode Convolutional Neural Network (CNN), implementasi CNN menggunakan Tensorflow dengan bahasa pemrograman python. Penelitian ini dilakukan dengan menggunakan 5 klasifikasi dataset tipe tas berjumlah 6.720 citra yang telah dilatih dengan ukuran citra 180 x 180 menggunakan kamera 360°. Diharapkan sistem ini mampu bekerja dengan baik untuk mengklasifikasikan jenis tas dalam format citra 360° (fish eye). Penelitian ini menghasilkan true detection rate sebesar 55% dan false detection sebesar 45% dimana true detection dilihat dari jumlah kebenaran akurasi yang menentukan hasil keluaran, sedangkan false detection merupakan kebalikan dari true detection dari jumlah 135 citra yang dihasilkan telah diuji.*

Kata Kunci— *Deep Learning, Convolutional Neural Network, Tensorflow, Kamera 360°, Jenis Tas, Klasifikasi.*

I. PENDAHULUAN

Perkembangan teknologi saat ini bergerak cukup pesat, sehingga memungkinkan untuk dapat diaplikasikan diberbagai bidang kehidupan. Khususnya pada bidang ilmu pengetahuan dan sistem komunikasi. Dampak yang ditimbulkan sampai kepada dunia *fashion* yaitu dimana wanita maupun pria remaja sampai orang tua masa kini tidak lepas dari tas sebagai tambahan *fashion* sehari-hari. Klasifikasi gambar merupakan salah satu metode *Machine Learning / Artificial Intelligence* yang dapat digunakan untuk mendeteksi sebuah gambar dengan cepat. Dalam pemilihan tas seringkali mengalami kesalahan, agar tidak salah pemilihan maka dalam tugas akhir saya ini melakukan klasifikasi jenis tas merupakan cara yang digunakan untuk membedakan tas berdasarkan jenis tas seseorang. Sistem klasifikasi tas berdasarkan jenisnya merupakan program yang telah dilakukan training dan disimpan pada database program yang dijalankan.

Klasifikasi jenis tas dapat dilakukan dengan berbagai cara, salah satunya *Deep Learning* dengan metode *Convolutional Neural Network* (CNN), implementasi CNN menggunakan *Tensorflow* dengan bahasa pemrograman *python*.

Convolutional Neural Network (CNN) adalah algoritma *Deep Learning* yang dirancang khusus untuk bekerja dengan gambar dan video. Dibutuhkan gambar sebagai input, mengekstrak dan mempelajari fitur gambar, dan mengklasifikasikannya berdasarkan fitur yang dipelajari. Algoritma ini terinspirasi oleh cara kerja bagian otak manusia yaitu *Visual Cortex*. *Visual Cortex* adalah bagian dari otak manusia yang bertanggung jawab untuk memproses informasi visual dari dunia luar serta memiliki lapisan yang berbeda dan setiap lapisan memiliki fungsinya sendiri yaitu setiap lapisan mengekstrak beberapa informasi dari setiap gambar atau visual apapun dan akhirnya semua informasi yang didapat dari setiap lapisan digabungkan, dan gambar/visual tersebut ditafsirkan atau diklasifikasikan. Demikian juga, CNN mempunyai berbagai filter, dan setiap filter mengekstrak beberapa informasi dari gambar dan kemudian semua itu digabungkan untuk mengidentifikasi gambar. Pada penelitian ini akan dibangun konstruksi CNN untuk program pendeteksian jenis tas pada gambar 360° (*fish eye*). Model algoritmanya akan diimplementasikan pada komputer dan informasi gambar diperoleh dari kamera 360°. Diharapkan sistem pendeteksian ini mampu bekerja baik mendeteksi jenis tas dalam format gambar 360° yaitu jpg.

II. LANDASAN TEORI

Convolutional Neural Network adalah salah satu algoritma *Deep Learning* yang merupakan pengembangan dari *Multilayer Peceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara. CNN dibuat dengan prinsip *translation invariance* yaitu dapat mengenali objek dalam citra pada berbagai macam posisi yang mungkin. Terdapat 2000 citra daun yang diklasifikasi menggunakan Alexnet. Alexnet merupakan arsitektur CNN milik Krizhevsky yang memiliki delapan layer ekstraksi fitur. Layer tersebut terdiri dari lima layer konvolusi dan tiga *pooling layer*. Dalam layer klasifikasinya, Alexnet mempunyai *dua layer Fully Connected* yang masing-masing mempunyai 4096 neuron. Pada akhir layer terdapat pengklasifikasian kedalam 20 kategori menggunakan aktifasi softmax. Rata-rata akurasi dari hasil klasifikasi mencapai 85%. Sedangkan akurasi dari identifikasi berhasil mencapai 90% yang didapatkan dari pengujian 40 citra[1].

Pada penelitian ini, dilakukan pengujian dengan mengklasifikasikan gambar wajah yang menggunakan masker dan tanpa menggunakan masker dengan menggunakan algoritma *Convolutional Neural Network* (CNN). Hasil penelitian ini menunjukkan model yang dibangun dapat melakukan klasifikasi gambar dengan tingkat akurasi mencapai 96%. Pengujian pada gambar wajah yang menggunakan masker memperoleh nilai precision 98%, recall 94% dan gambar wajah yang tidak menggunakan masker memperoleh nilai precision 94%, recall 98[2].

A. Artificial Intelligent

Kecerdasan buatan atau *Artificial Intelligence* ialah satu dari beberapa cabang ilmu komputer yang mempelajari bagaimana melatih suatu mesin agar bisa melakukan sebuah pekerjaan yang apik dan rapi selayaknya manusia, atau bahkan dapat melakukan lebih baik dari manusia[3].

Kecerdasan buatan adalah kecerdasan yang ditambahkan oleh manusia ke dalam suatu sistem teknologi, diatur dan dikembangkan dalam konteks ilmiah, bentukan dari kecerdasan entitas ilmiah yang ada. Pada dasarnya AI adalah suatu pengetahuan yang membuat komputer dapat meniru kecerdasan manusia sehingga komputer dapat melakukan hal-hal yang dikerjakan manusia dimana membutuhkan suatu kecerdasan; misalkan melakukan analisa penalaran untuk mengambil suatu kesimpulan atau keputusan atau penerjemahan dari satu bahasa ke bahasa lain.

B. Deep Learning

Deep Learning adalah salah satu teknik pada *machine learning* yang memanfaatkan banyak *layer* pengolahan informasi *non-linier* untuk melakukan ekstrasi fitur, pengenalan pola, dan klasifikasi. *Deep learning* adalah salah satu cabang dari *machine learning* untuk penyelesaian masalah dengan dataset yang cukup besar menggunakan jaringan syaraf tiruan. Metode ini memiliki arsitektur yang amat kuat untuk *15 supervised learning*. *Supervised Learning* yaitu tipe pembelajaran yang berisi variabel masukan dan variabel keluaran dengan menggunakan satu algoritma atau lebih untuk mempelajari fungsi penggambaran dari masukan ke keluaran. Dengan lapisan yang ditambahkan lebih banyak maka model pembelajaran bisa mewakili data citra berlabel dengan lebih baik. Selain itu, program *machine learning* pada umumnya hanya menggunakan CPU dan RAM dalam proses komputasi, yang menentukan kecepatan dalam mengkomputasi data adalah spesifikasi dari CPU dan RAM. Selain membutuhkan CPU dan RAM pada proses komputasi, kemampuan GPU juga diperlukan pada proses ini agar dalam pemrosesan data yang besar dapat berlangsung dengan cepat [4].

Convolutional Neural Network (CNN) merupakan pengembangan dari *multilayer perceptron (MPL)* yang didesain untuk mengolah data dua dimensi dalam bentuk citra. CNN ini termasuk kedalam jenis *Deep Neural Network* karena kedalaman jaringan yang tinggi dan banyak diaplikasikan pada data citra. CNN adalah sebuah arsitektur yang dapat dilatih dan terdiri dari beberapa tahap, Masukan dan keluaran dari setiap tahap itu terdiri dari beberapa *array* yang biasa disebut *feature map*. Setiap tahap terdiri dari tiga *layer* yaitu konvolusi, fungsi aktivasi *layer* dan *pooling layer*.

III. METODE/MODEL YANG DIUSULKAN

A. Metodologi Penelitian

Metodologi penelitian ini melakukan penelitian secara sistematis yang dimulai dari tahapan studi literatur sampai pembuatan laporan tugas akhir ini. Tahapan, metode, dan hasil penelitian akan diperlihatkan pada Tabel 1 dibawah ini.

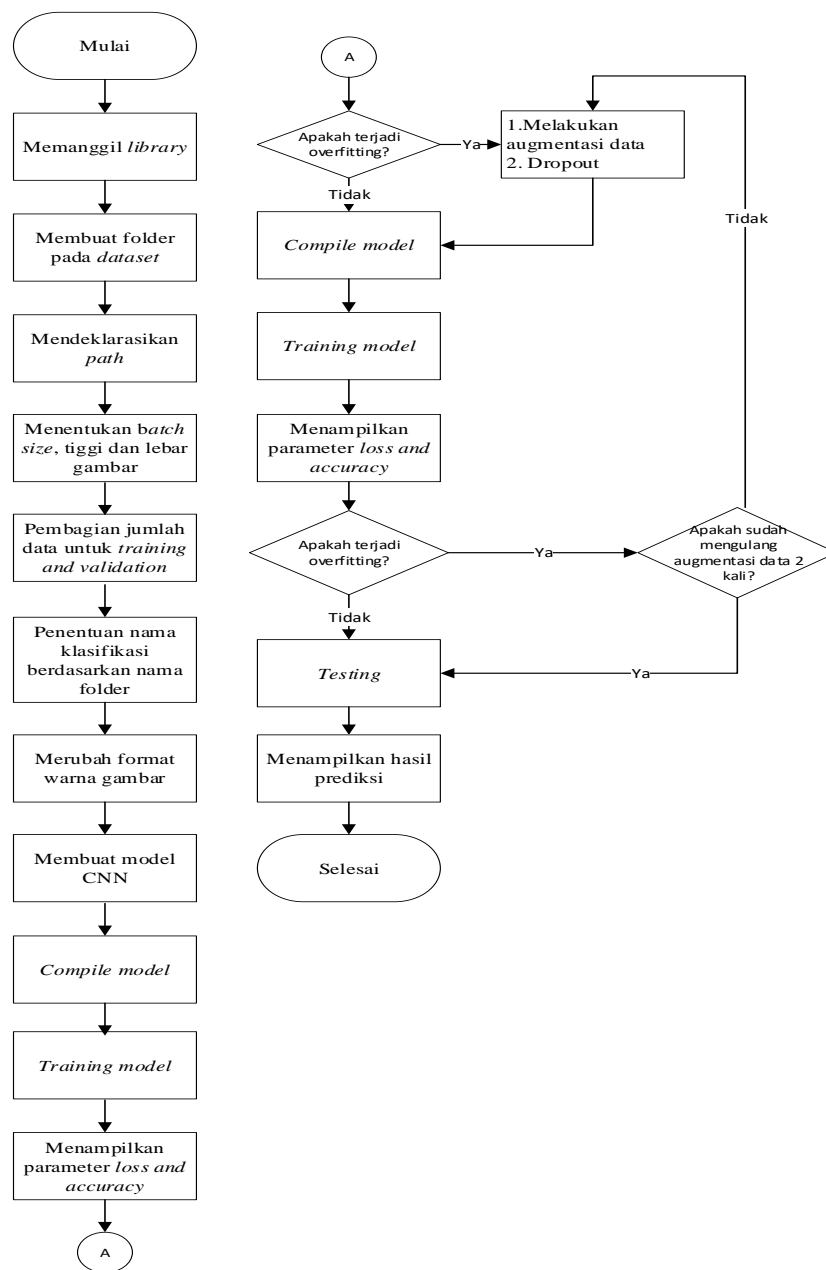
Tabel 1. Tahapan Penelitian

No	Tahapan	Metode	Hasil
1.	Studi kepustakaan	1. Peninjauan internet 2. Studi buku serta jurnal terkait	Pemahaman: 1. Mengenai teknologi <i>Convolutional Neural Network (CNN)</i> dan pemrosesannya 2. Pemrograman dengan bahasa <i>Python</i> 3. <i>Hardware</i> dan <i>software</i> yang digunakan
2.	Desain perangkat lunak klasifikasi jenis tas pada gambar 360°	Menggunakan <i>software</i> pemograman <i>python</i>	Desain <i>software</i> pengolahan data pada CNN
3.	Pengumpulan <i>dataset</i>	1. <i>Dataset</i> diambil menggunakan kamera 360° 2. Mengubah video ke <i>frame</i>	<i>Dataset</i> segmentasi dengan jumlah 6720 <i>frame</i> siap uji dengan keterangan klasifikasi

No	Tahapan	Metode	Hasil
		3. Membuat klasifikasi data	
4.	Realisasi perangkat lunak pada PC	Memakai <i>python</i> sebagai <i>software</i> pada pemrogramannya	<i>Software</i> menerima dan menyimpan data
5.	Pengukuran kinerja <i>software</i> melakukan analisis	1. Pengukuran skala laboratorium 2. Analisis kuantitatif 3. Analisis kualitatif	Kinerja <i>software</i> dalam melatih dan menguji data, serta kesimpulan penelitian

B. Flowchart

Gambar 1 di bawah ini merupakan langkah dan alur klasifikasi objek pada penelitian ini.

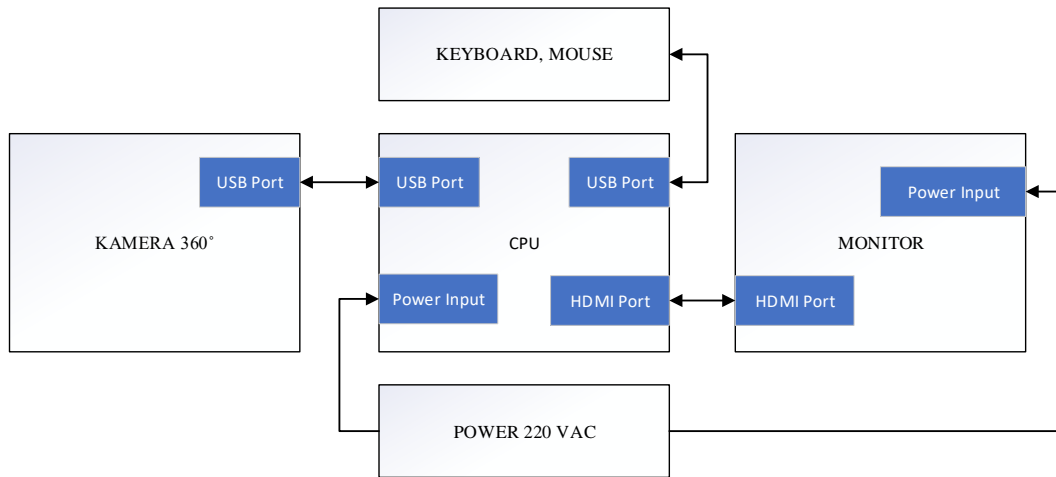


Gambar 1 Flowchart Sistem Klasifikasi Jenis Tas.

IV. HASIL/IMPLEMENTASI MODEL DAN PEMBAHASAN

Bagian keempat ini menjelaskan perihal desain hingga pengaplikasian klasifikasi jenis tas secara rinci, sesuai seperti klasifikasinya di area terbatas yang berdasar *Deep Learning* menggunakan arsitektur CNN menggunakan gambar dari kamera 360°.

A. Desain Model




Gambar 2. Ilustrasi Hardware





Gambar 1 diatas menunjukkan desain Gambar *Hardware* yang digunakan pada klasifikasi jenis tas pada gambar 360° (*fish eye*) dengan menggunakan *Tensorflow*.

B. Pembahasan

Tahapan ini menjelaskan tahapan analisis pada klasifikasi jenis tas pada gambar 360° (*fish eye*) dengan menggunakan *Tensorflow*. Pada penelitian ini pengujian terhadap jenis tas, klasifikasinya berupa 5 jenis tas berbeda. Hasil pengujian yang diambil dalam penelitian ini ialah ketepatan serta nilai keakuratan dalam pengklasifikasian jenis tas yang sudah dibuat dengan baik terhadap 6720 gambar secara *offline*. Dataset yang diambil pada klasifikasi jenis tas dapat dilihat pada Tabel 2.

Tabel 2. Klasifikasi Dataset

No	Nama Klasifikasi	Jumlah Data	Gambar
1.	<i>Waist_bag</i>	1.290	

No	Nama Klasifikasi	Jumlah Data	Gambar
2	<i>Tote_bag</i>	1.350	
3.	<i>Sling_bag</i>	1.470	
4.	<i>Round_rattan_bag</i>	1.290	
5.	<i>Backpack</i>	1.320	

Tabel 2 menunjukkan *dataset* yang diambil untuk nantinya akan diuji pada proses *training dataset*. *Output* dari *training dataset* berupa gambar yang telah diuji apakah gambar tersebut tepat atau tidak sesuai hasil *percent confidence*. Dalam hal ini terdapat *training and validation*. Dimana *Training dataset* merupakan himpunan data yang digunakan untuk melatih atau membangun model. Kemudian, *validation dataset* merupakan himpunan data yang digunakan untuk mengoptimasi saat melatih model. Model dilatih menggunakan *training dataset*, kemudian kinerja saat latihan tersebut diuji menggunakan *validation dataset*. Hal ini bertujuan untuk melihat kemampuan model pada saat *training* apakah dapat mengenal pola secara umum. Pengujian ini dibagi menjadi 3, berdasarkan epoch yang digunakan yaitu, *epoch* (10, 15) (20, 30) (30, 45) serta *training and validation* 0.1 (90% *Training*, 10% *Validation*), 0.2 (80% *Training*, 20% *Validation*), 0.3 (70% *Training*, 30% *Validation*) dan menguji gambar. Pengujian tersebut dipilih satu yang paling tepat dan akan dijelaskan dibawah ini.


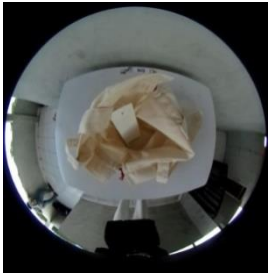

Pengujian ini menggunakan epoch (20, 30) dan hasil dari pembagian dataset menggunakan *training and validation* 0.1 (90% Training, 10% Validation), serta menguji 15 gambar, yang ditunjukkan Tabel III





Tabel 3. Hasil *Training and Validation* 0.1 epoch (20, 30)





Epoch	Hasil
Pertama (20)	
Kedua (30)	





Pada pengujian ini menggunakan 15 gambar uji yaitu 4 gambar uji menggunakan dataset baru, 8 gambar uji menggunakan dataset yang ditraining dan 3 gambar uji random. Pada tabel IV diperlihatkan hasil pengujian tersebut.

Tabel 4. Hasil Uji Prediksi

No	Gambar Output	Hasil prediksi	Ket
1.		<pre>Predict on new data Use your model to classify an image that wasn't included in the training or validation sets. bag_path = "C:/file_agnes/Waist_bag/Waist_bag_cut/Waist_bag_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) 1/1 [=====] - 0s 99ms/step This image most likely belongs to waistbag with a 100.00 percent confidence.</pre>	Tepat
2.		<pre>Predict on new data Use your model to classify an image that wasn't included in the training or validation sets. bag_path = "C:/file_agnes/Tote_bag/Tote_bag_cut/Tote_bag_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) 1/1 [=====] - 0s 15ms/step This image most likely belongs to totebag with a 100.00 percent confidence.</pre>	Tepat
3		<pre>Predict on new data Use your model to classify an image that wasn't included in the training or validation sets. bag_path = "C:/file_agnes/Sling_bag/Sling_bag_cut/Sling_bag_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) 1/1 [=====] - 0s 16ms/step This image most likely belongs to slingbag with a 99.48 percent confidence.</pre>	Tepat

No	Gambar Output	Hasil prediksi	Ket
4		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/Backpack/Backpack_cut/Backpack_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to backpack with a 100.00 percent confidence.</p>	Tepat
5		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/new_file_Agnes/Tote_bag/Tote_bag_cut/Tote_bag_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to waistbag with a 100.00 percent confidence.</p>	Tidak Tepat
6.		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/Waist_bag/Waist_bag_cut/Waist_bag_cut_100.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to waistbag with a 99.98 percent confidence.</p>	Tepat
7		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/Tote_bag/Tote_bag_cut/Tote_bag_cut_200.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to backpack with a 99.65 percent confidence.</p>	Tidak Tepat

No	Gambar Output	Hasil prediksi	Ket
8		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/Sling_bag/Sling_bag_cut/Sling_bag_cut_200.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 15ms/step This image most likely belongs to waistbag with a 74.64 percent confidence.</p>	Tidak Tepat
9		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/Backpack/Backpack_cut/Backpack_cut_180.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to backpack with a 99.86 percent confidence.</p>	Tepat
10		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/new_file_Agnes/Tote_bag/Tote_bag_cut/Tote_bag_cut_100.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to waistbag with a 100.00 percent confidence.</p>	Tidak Tepat
11		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/new_file_Agnes/Sport_bag/Sport_bag_cut/Sport_bag_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to waistbag with a 100.00 percent confidence.</p>	Tidak Tepat

No	Gambar Output	Hasil prediksi	Ket
12		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_tejo/aris_cut/aris_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to waistbag with a 100.00 percent confidence.</p>	Tidak Tepat
13		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_devit/Sereh_Cut/Sereh_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to roundbag with a 99.86 percent confidence.</p>	Tidak Tepat
14		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/new_file_Agnes/Waist_bag/Waist_bag_cut/Waist_bag_cut_1.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 16ms/step This image most likely belongs to waistbag with a 100.00 percent confidence.</p>	Tepat
15		<p>Predict on new data</p> <p>Use your model to classify an image that wasn't included in the training or validation sets.</p> <pre> bag_path = "C:/file_agnes/new_file_Agnes/Waist_bag/Waist_bag_cut/Waist_bag_cut_70.jpg" img = tf.keras.utils.load_img(bag_path, target_size=(img_height, img_width)) img_array = tf.keras.utils.img_to_array(img) img_array = tf.expand_dims(img_array, 0) # Create a batch predictions = model.predict(img_array) score = tf.nn.softmax(predictions[0]) print("This image most likely belongs to {} with a {:.2f} percent confidence." .format(class_names[np.argmax(score)], 100 * np.max(score))) </pre> <p>1/1 [=====] - 0s 15ms/step This image most likely belongs to waistbag with a 100.00 percent confidence.</p>	Tepat

Pada Tabel 4 hasil pengujian dengan menggunakan *epoch* (20, 30) dengan *training and validation* 0.1 didapatkan hasil pengujian *epoch* pertama (20) tidak terjadi fluktuasi yang signifikan namun pada *epoch* kedua (30) terjadi fluktuasi. Pengujian prediksi gambar pada *training and validation* 0.1 dari 15 gambar yang diuji terdapat 8 gambar yang menunjukkan klasifikasi tepat dan 7 gambar menunjukkan klasifikasi tidak tepat.

V. KESIMPULAN

Bagian kelima ini adalah kesimpulan dari klasifikasi jenis tas pada gambar 360° (*fish eye*) dengan menggunakan *Tensorflow*. Adapun kesimpulan yang didapatkan dari hasil penelitian ini adalah sebagai berikut:

1. Klasifikasi jenis tas berhasil dilakukan dengan presentase 55% *true detection* dan 45% *false detection* dimana *true detection* dilihat dari banyaknya kebenaran ketepatan dalam menentukan hasil *output*, sedangkan *false detection* merupakan kebalikan dari *true detection* dari jumlah 135 gambar yang telah diujikan.
2. Jumlah dataset mempengaruhi keakuratan pada proses klasifikasi.
3. Semakin banyak jumlah epoch yang digunakan maka semakin lama waktu yang dibutuhkan tetapi menghasilkan klasifikasi yang lebih akurat, sedangkan semakin sedikit jumlah epoch yang digunakan maka tidak terlalu banyak waktu yang digunakan akan tetapi menghasilkan klasifikasi yang kurang akurat.

UCAPAN TERIMA KASIH

Ucapan terima kasih kami sampaikan kepada pihak-pihak yang telah berkontribusi dalam penelitian ini hingga makalah ini dapat dipublikasikan pada Seminar Nasional Sains Teknologi dan Inovasi Indonesia 2022, diantaranya kepada:

1. LPDP (Lembaga Pengelola Dana Pendidikan) sebagai sponsor penelitian.
 2. BRIN (Badan Riset dan Inovasi Nasional) sebagai penyedia prasarana penelitian.
 3. Universitas Nurtanio sebagai penyedia prasarana penelitian.
 4. AAU (Akademi Angkatan Udara) sebagai penyelenggara konferensi dan prosiding.
- Serta rekan-rekan –rekan peneliti lainnya.

REFERENSI

- [1] Ilahiyah, S., & Nilogiri, A. (2018). Implementasi *Deep Learning* Pada Identifikasi Jenis Tumbuhan Berdasarkan Citra Daun Menggunakan *Convolutional Neural Network*. JUSTINDO (Jurnal Sistem Dan Teknologi Informasi Indonesia), 3(2), 49-56.
- [2] Rahim, A., Kusriani, K., & Luthfi, E. T. (2020). *Convolutional Neural Network* untuk Kalasifikasi Penggunaan Masker. *Inspiration: Jurnal Teknologi Informasi dan Komunikasi*, 10(2), 109-115.
- [3] Athanasia Octaviani Puspita Dewi (2020), Kecerdasan Buatan sebagai Konsep Baru pada Perpustakaan
- [4] Triano Nurhikmat (2018). Implementasi *Deep Learning* untuk *Image Classification* Menggunakan Algoritma *Convolutional Neural Network* (CNN) pada Citra Wayang Golek.