

TRANSFER LEARNING TO PREDICT GENRE BASED ON ANIME POSTERS

Kaka Kamaludin^{*1}, Woro Isti Rahayu², Muhammad Yusril Helmi Setywan³

^{1,2,3}Informatics Engineering Study Program, Universitas Logistik dan Bisnis Internasional, Indonesia
Email: ¹kaka_kamaludin666@hotmail.com, ²woroisti@ulbi.ac.id, ³yusrilhelmi@ulbi.ac.id

(Article received: January 26, 2023; Revision: March 12, 2023; published: October 15, 2023)

Abstract

Anime is an animated film with a distinctive graphic design originating from Japan, which is widely favored by various groups. anime itself has a genre like a movie in general, but there is a slight difference from ordinary films, anime has additional genres that are not in ordinary films, such as the *Ecchi*, *Mahou Shoujou*, *Seinen*, *Shounen*, and *Josei* genres. Since those genres only exist in anime, this research is devoted to predicting those anime genres. The prediction will use posters from the anime itself, with the help of image processing, namely the Convolutional Neural Network method and Transfer Learning. Transfer Learning will be implanted as a comparison of the performance of the existing architecture with the architecture that will be created, whether the architecture is able to process the dataset properly. The dataset to be used is a dataset of posters and csv documents containing images and details of the anime, the dataset contains anime data from 1980 to 2021 and contains 11651 anime poster data which has different resolution sizes. The ResNet50 model has the highest accuracy rate of 48% with a loss rate of 36%, while InceptionV3 produces 35% accuracy with 69% loss. At the time of testing ResNet50 gave the smallest genre percentage value of CustomModel and InceptionV3, while CustomModel gave the highest genre value. In addition to the value, all modes also predicted the genre well. Especially InceptionV3 is able to predict the music genre, because the music genre has a very small number of datasets, and this music genre is difficult to predict by the ResNet50 and CustomModel models.

Keywords: anime, genre, neural network, poster, transfer learning.

PREDIKSI GENRE ANIME PADA POSTER DENGAN METODE TRANSFER LEARNING

Abstrak

Anime merupakan film animasi dengan desain grafis khas berasal dari Jepang, yang banyak digemari berbagai kalangan. anime sendiri memiliki genre seperti film pada umumnya, namun ada sedikit perbedaan dari film biasa, anime memiliki genre tambahan yang tidak ada pada film biasa, seperti genre *Ecchi*, *Mahou Shoujou*, *Seinen*, *Shounen*, dan *Josei*. Karena genre-genre itu hanya ada pada anime, maka penelitian ini dikhususkan untuk memprediksi genre-genre anime tersebut. Prediksi tersebut akan menggunakan poster dari anime itu sendiri, dengan bantuan *image processing* yaitu metode *Convolutional Neural Network* dan *Transfer Learning*. *Transfer Learning* akan diimplantasikan sebagai perbandingan dari kinerja arsitektur yang sudah ada dengan arsitektur yang akan dibuat, apakah arsitektur tersebut mampu memproses *dataset* dengan baik. *Dataset* yang akan digunakan adalah *dataset* poster dan dokumen *csv* yang berisi gambar dan detail dari animenya, *dataset* berisi data anime dari tahun 1980 sampai 2021 dan berisi 11651 data poster anime yang memiliki ukuran resolusi yang berbeda. Model ResNet50 memiliki tingkat akurasi paling tinggi sebesar 48% dengan tingkat *loss* 36%, sedangkan InceptionV3 menghasilkan akurasi 35% dengan *loss* 69%. Pada saat pengujian *ResNet50* memberikan nilai persentase genre paling kecil dari *CustomModel* dan *InceptionV3*, sedangkan *CustomModel* memberikan nilai genre paling tinggi. Selain nilai, semua mode juga berhasil memprediksi genre dengan baik. Terutama *InceptionV3* yang mampu memprediksi genre *music*, karena genre *music* memiliki jumlah *dataset* yang sangat sedikit, dan genre *music* ini sulit di prediksi oleh model *ResNet50* dan *CustomModel*.

Kata kunci: anime, genre, neural network, poster, transfer learning.

1. PENDAHULUAN

Setiap orang atau individu tentu dan pastinya memiliki hobinya masing-masing, seperti halnya

otaku anime. Menurut KBBI hobi adalah kegemaran atau kesenangan istimewa pada waktu senggang, bukan pekerjaan utama. *Otaku* adalah julukan yang diberikan kepada orang yang memiliki ketertarikan

kepada budaya populer Jepang seperti anime, *manga*, *Jpop*, dan *tokusatsu* [1]. dan Anime adalah istilah untuk menyebut film animasi atau kartun, animasi budaya pop Jepang. Kata anime merupakan singkatan dari “*animation*” dalam bahasa Inggris, yang merujuk pada semua jenis animasi [2], [3]. Seperti halnya film biasa, pada anime juga terdapat poster, poster memiliki tujuan utama menarik perhatian, menginformasikan, meyakinkan dan memprovokasi. Selain itu poster film merupakan sarana yang efektif dalam komunikasi visual dan fungsi komersial [4][5]. Poster anime sendiri cenderung menggambarkan isi dari anime itu sendiri, seperti genre *Romance* yang posternya selalu ada gambar pasangan kekasih, lalu *Comedy* yang posternya berisi orang tertawa atau lucu, dan yang lebih gampang lagi adalah genre *War* atau *Mecha* yang sudah pasti terdapat gambar senjata dan robot. Karena desain poster anime yang khas dengan *art style* berbeda, terkadang sulit untuk membedakan genre dari film anime tersebut.

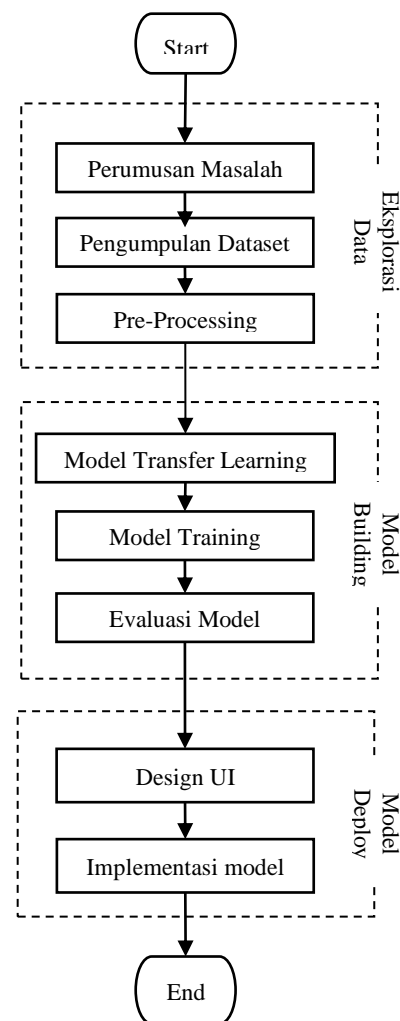
Berdasarkan paparan pada paragraf di atas maka dibuatlah aplikasi prediksi genre anime pada gambar poster. Penelitian sejenis mengenai genre berdasarkan gambar sudah dilakukan, dengan tema Klasifikasi Genre Manga Menggunakan *Convolutional Neural Network*. Penelitian tersebut menggunakan *multi-label classification* dengan jumlah 25 genre yang di klasifikasi, 10.000 *dataset training*, 9.000 data testing yang berisi gambar berwarna dan hitam putih. Model yang dibuat diuji sebanyak 39 kali, 25 genre berhasil diprediksi dengan tingkat akurasi 46.3% [6]. Keberhasilannya Metode CNN dalam prediksi tersebut sangat menarik untuk di terapkan pada prediksi genre anime, karena objek gambar yang dipakai memiliki karakteristik yang sama-sama khas kartun jepang.

Penelitian prediksi genre anime akan di buat menggunakan teknik klasifikasi *multi-class classification* dengan model *InceptionV3* dan *ResNet50*. Selain dua model tersebut *Custom Model* juga akan di kembangkan kembali dengan referensi dari model *InceptionV3* dan *ResNet50*. *multi-class classification* adalah klasifikasi sebuah objek ke satu atau lebih kelas [7]. Arsitektur *ResNet-50* adalah Jaringan *Residual* yang terdiri dari 50 layer. Seperti yang ditunjukkan oleh namanya, *network* ini menggunakan pembelajaran *Residual*, dan *Resnet50* yang merupakan model yang sudah dilatih menggunakan *ImageNet Dataset* agar dapat menghasilkan *feature extraction* yang baik. [8][9]. *Inceptionv3* juga dikenal sebagai *GoogLeNet* adalah varian dari model *deep learning* yang dibangun oleh Google dan versi ke-3 CNN dari keluarga arsitektur *Inception* yang mengalami beberapa perbaikan. Selain itu, *InceptionV3* menggunakan pengklasifikasi tambahan untuk memperluas informasi label ke bawah jaringan dan membantu lebih lanjut dalam regularisasi [10]–[12]. *Dataset* yang digunakan berupa poster anime dan *file CSV* berisi informasi dari setiap poster, seperti judul, genre, dan tahun.

Dataset ini berjumlah 11651 yang di kumpulkan dengan cara *scraping*, poster yang digunakan adalah data dari tahun 2013 sampai 2021. Implementasi dari penelitian ini akan menggunakan web *Framework* yang bernama *Streamlite*. *Streamlit* adalah *library Python open-source* dan gratis untuk pengembangan aplikasi web, Karena kode-kodenya juga tersedia untuk umum di *GitHub*, para developer dapat dengan mudah menyesuaikan kode-kode tersebut untuk mengubah dan memperbarui apa pun yang diinginkan [13][14]. *Streamlit* digunakan karena kemampuannya dalam membuat *dashboard* dengan cepat. Tujuan dan capaian dari pembahasan ini adalah membuat aplikasi yang mampu memprediksi genre anime dengan tingkat akurasi yang baik.

2. METODE PENELITIAN

Diagram alur metodologi penelitian yang digambarkan pada *Gambar 1*. Dibuat berurutan dan dikelompokkan dari satu tahap ke tahap lainnya. Berikut penjelasan dari tahapan-tahapan pada diagram alur metodologi penelitian



Gambar 1. Diagram alur metodologi penelitian

2.1. Perumusan Masalah

Pada tahap awal ini, penelitian membahas tentang apa saja yang perlu dipersiapkan untuk kebutuhan penelitian. Hal yang pertama dibahas adalah topik dari penelitian itu sendiri yaitu prediksi genre anime, kemudian setelah mempunyai topik, tahap selanjutnya adalah membahas bagaimana tatanan cara agar dapat melakukan prediksi genre anime. Di penelitian ini akan menggunakan metode *Transfer Learning* untuk meraih tujuan dari penelitian. Kemudian membahas juga tentang dari mana data akan diambil, jumlah dari datanya itu sendiri dan hal-hal lainnya yang berkaitan dengan penelitian.

2.2. Pengumpulan Dataset

Tahap pengumpulan *data set* ini adalah tahap untuk mengumpulkan data yang dibutuhkan untuk penelitian. Data yang digunakan akan didapatkan dari berbagai sumber. Data ini berbentuk *csv* dan poster *Gambar 2*, serta terdapat beberapa *column* yang nantinya akan digunakan untuk dilakukan prediksi

yaitu *column* dari setiap genre, *year* dan lain-lain. data *year* yang akan digunakan ini akan berada dalam jangka waktu sepuluh tahun terakhir dari tahun 2021. *Dataset* inilah yang akan digunakan untuk penelitian dalam memprediksi genre *anime*. Contoh pada Tabel 1. dan Tabel 2.



Gambar 2. Poster Anime

Tabel 1. *Dataset Anime – part 1*

	Image	Year	Genres	Rating	Type	Format	Studios
3726	Poster/Coppelion.jpg	2013	[Action, Sci-Fi]	60	ANIME	TV	['GoHands']
10612			[Action, Adventure, Sci-Fi]				['Tatsunoko Production']
10613	Poster/GatchamanCrowds.jpg	2013	[Action, Drama, Supernatural]	71	ANIME	TV	['Studio DEEN']
10614	Poster/HAKUOKIWILDDAN CEOFTOKYO.jpg	2013	[Drama, Romance, Sci-Fi]	70	ANIME	MOVIE	['Wit Studio']
10616	Poster/Hal.jpg	2013	[Action, Adventure, Fantasy]	70	ANIME	MOVIE	['MADHOUSE']

Tabel 2. *Dataset Anime – part 2*

Act ion	Adv entu re	Co me dy	Dra ma	Ecc hi	Fan tasy	Ho rror	Mahou Shoujo	M eca ha	M usic	Myst ery	Psyc holo gical	Ro ma nce	Sci -Fi	Sp ort s	Slice of Life	Sup e rnat ural	Th riller
1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0	0
1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

2.3. Pre-Processing

Tahap *preprocessing* adalah tahapan dimana dilakukan seleksi data agar data yang akan digunakan menjadi lebih terstruktur [15]. Tahap *Pre-processing* ini akan menghasilkan data yang siap di olah, dengan cara menyeleksi dan membersihkan data kosong, ini dilakukan dengan tujuan data tidak menghambat proses *Transfer Learning*. Proses yang terjadi di tahap ini adalah penyesuaian *dataset csv*, *dataset csv* akan dilakukan *not-null filter* dan *year range*. Sedangkan data Gambar Poster akan di terapkan *data augmentation*, yang bertujuan untuk menyelaraskan resolusi gambar.

2.4. Model Transfer Learning

Pada tahap ini, rancangan model yang akan menggunakan *weight* model *ResNet50* dan *InceptionV3*. *Weight ImageNet* akan digunakan

sebagai pembobotan dalam pembuatan model, Proses tersebut dilakukan dengan menggunakan *keras API*. *ImageNet* adalah tempat penyimpanan *dataset* gambar yang sangat besar dan memiliki kompetisi yang terbuka setiap tahun untuk mempromosikan penelitian. Pada 2012, pemenang kompetisi *ImageNet* adalah model *Alexnet* oleh *Alex Krizhevsky* [16]. setelah berhasil mendapatkan *weight* dan merancangny menjadi model *Transfer Learning*, maka *dataset* yang sudah di *pre-processing* akan muat ke dalam model ini, untuk dilakukan model *training*.

2.5. Model Training

Pelatihan model merupakan proses pelatihan mengenali objek dan mengklasifikasikannya sesuai dengan kelasnya [17]. Setelah rancangan model selesai di buat dan optimasi yang ideal di temukan.

Sebelum melatih model, diperlukan *parameter optimization*. *parameter* yang harus di tentukan adalah jumlah *epoch* atau jumlah berapa kali model akan belajar dengan muatan *data training*. Proses *training* adalah tahapan dimana CNN dilatih untuk memperoleh akurasi yang lebih tinggi dari klasifikasi yang dilakukan. Pada tahapan ini terdapat proses *feed forward* dan proses *backpropagation*. [18]. Proses ini akan terus dilakukan sampai mendapat hasil prediksi dan akurasi dari *data training* yang ideal. *Feed forward* merupakan proses untuk mengetahui *output* dari jaringan syaraf tiruan. *Backpropagation* merupakan proses untuk memperbarui nilai bobot dan nilai bias jika nilai *output* dari perhitungan *feedforward* belum sesuai dengan *output* dari target sebenarnya atau nilai eror dari *output* yang dihasilkan tidak lebih kecil atau sama dengan toleransi eror [19].

2.6. Evaluasi Model

Proses evaluasi model adalah proses pengujian yang dilakukan untuk mengetahui kinerja suatu model apabila diaplikasikan pada kasus yang belum pernah digunakan dalam proses pelatihan (unseen data). Proses evaluasi sangat penting untuk mengetahui bahwa suatu model memiliki kinerja yang baik tidak hanya pada suatu himpunan data tertentu [20]. Pada tahap ini hasil *training Accuracy* dan *Loss* akan di kaji, setiap nilai dari kedua *function* tersebut akan menjadi acuan apakah model yang dipakai mampu memproses *dataset*. Persentase dari *Accuracy* dan *Loss* setiap modelnya akan dibandingkan untuk menentukan *model* mana yang optimal.

2.7. Design UI

Design UI merupakan tahap pembuatan tampilan untuk aplikasi web prediksi anime, perancangan ini akan dibuat menggunakan *wireframe*, yang berisikan elemen yang akan menjadi fitur *website*. *Wireframe* merupakan rancangan *user interface*, sistem yang akan dibangun dalam bentuk *mock up design* [21]. *User interface* merupakan serangkaian tampilan grafis yang dapat dimengerti oleh pengguna komputer dan diprogram sedemikian rupa sehingga dapat terbaca oleh sistem operasi komputer dan beroperasi sebagaimana mestinya. *User interface* adalah salah satu faktor yang menentukan peningkatan *traffic* pada sebuah *website* [22].

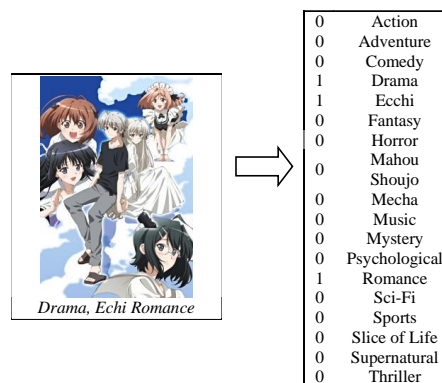
2.8. Implementasi Model

Tahap ini merupakan tahap akhir dari *machine learning*. Model yang telah selesai dibuat akan dipublikasikan dan diterapkan pada media Web. Model *Transfer Learning* akan di *import* ke *streamlit*, lalu akan dibuat fitur *Upload Image* sebagai masukan dari model, dan hasil prediksinya akan di tampilkan pada halaman depan aplikasi. Aplikasi yang akan di bangun berbasis web yang berbentuk *landing page*,

yang memuat semua fitur utama menjadi satu halaman.

3. HASIL DAN PEMBAHASAN

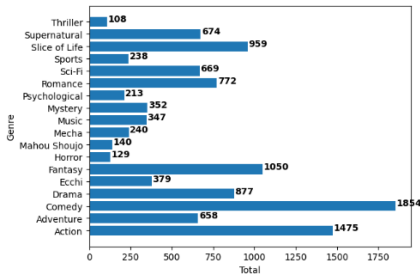
Dataset yang digunakan untuk melatih model terdiri dari dua bagian. Yang pertama adalah set poster anime dan yang lainnya, satu set label genre dari posternya. *Dataset* ini ambil dari web *anilist.co*, yang berisi *database* informasi anime termasuk judul anime, genre, *rating*, studio serta tahun rilisnya. *Database* ini berisi 11651 judul anime. Pada *database* ini berisi 18 genre. Poster nantinya akan dipakai untuk di ambil data RGB pada setiap layer untuk masing-masing gambar. RGB merupakan ruang warna yang dimiliki citra yaitu dengan tiga warna primer yaitu *red*, *green*, dan *blue*. Rentang nilai pada penskalaan RGB dalam setiap *pixel* dimulai dai skala 0 (nol) yaitu hitam hingga skala 255 (dua ratus lima puluh lima) yang mempunyai kedalaman warna yaitu dimulai dari 8 bit [23]. Data *pixel* dari poster yang ada nantinya akan diubah ukurannya menjadi 317 x 230 x 3. *Pixel* adalah elemen citra yang memiliki nilai yang menunjukkan intensitas warna [24]. Label genre yang sesuai akan di ubah menjadi *multi-hot vectors*, detail Gambar 3. *Multi-hot vectors* adalah vektor buatan yang dibuat pada proses *machine learning* untuk merepresentasikan *class* atau kategori klasifikasi dalam *multidimensional space* dengan mengkodekannya ke dalam nilai numerik.



Gambar 3. Genre anime di inisial kan sebagai *multi-hot vectors*

3.1. Pre-processing

Pre-processing yang dilakukan di sini adalah pemangkasan *dataframe* dan membersihkan *null value*. Salah satu pemangkasan yang dilakukan adalah dengan limitasi penyortiran *value* tahun rilis. Setelah penyortiran tahun rilis, maka selanjutnya adalah menghapus *cell* yang memiliki data *null*, pada *column* Rating, Studio dan Genre. Selain *column* itu terdapat juga *column Image* yang dilakukan normalisasi pada nama *file*, yaitu dengan penghapusan simbol. Hal yang sama juga dilakukan pada *column Genres*, yaitu penghapusan tanda kutip. *pre-processing* ini menyisakan data sebanyak 4040, dan Gambar 4 hasil dari total genre *dataset* yang telah di *pre-processing*.



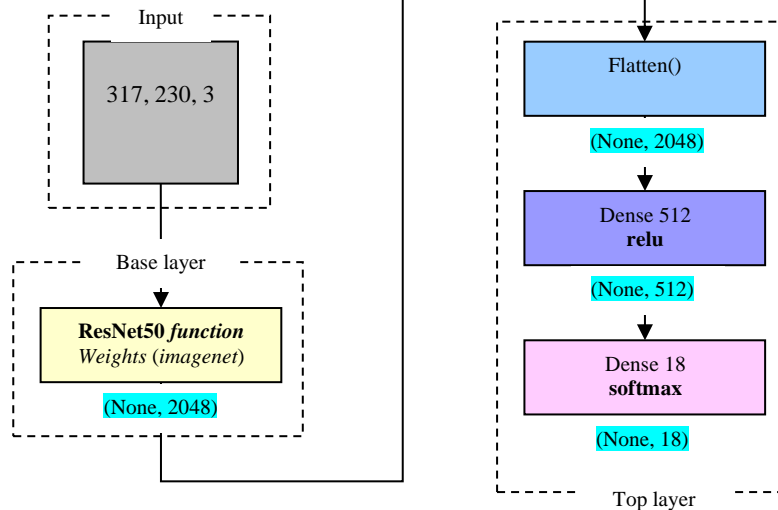
Gambar 4. Jumlah Genre yang akan dipakai.

3.2. Model ResNet50

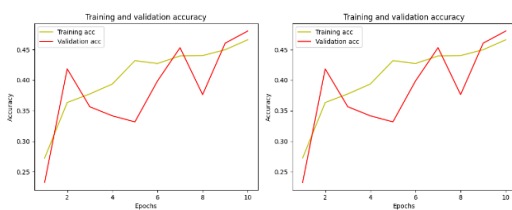
Untuk membangun model *neural network* adalah *initialitation* model dahulu, dengan menggunakan fungsi *Sequential()*. Lalu menyiapkan *pre-trained ResNet50 model* dari *Keras API*.

Weights Imagenet digunakan pada model ini, sebagai penambah bobot untuk model yang akan di

bangun. *input_shape* ukurannya disamakan dengan *dataset* gambar. Lalu untuk *include_top* ini tidak akan di pakai, model ini akan menggunakan *dense layer* sendiri untuk menyesuaikan dengan banyaknya *classes*. jumlah *classes* ini merupakan banyaknya genre yang di gunakan sebagai klasifikasinya. Dan *pooling* yang dipakai pada model ini adalah *MaxPooling*. Selain itu *weights* atau bobot dari *ResNet50* ini tidak akan melewati tahap *Model Training*, hal ini dilakukan untuk mengurangi waktu *training* karena terlalu banyaknya data. Pembuatan *layer* atas. Jumlah *hidden layer* atau *dense layer* yang digunakan disesuaikan dengan jumlah klasifikasi, dan menggunakan aktivasi *relu* dan *softmax*, detail arsitektur pada *Gambar 5*. Model *ResNet50* ini menggunakan *Adam Optimizer*, *loss* menggunakan *binary_crossentropy*, dan *metrics* menggunakan perhitungan *accuracy*. *training* menggunakan 10 *epoch*, detail *chart* pada *Gambar 6*.



Gambar 5. Flow model transfer learning Resnet50



Gambar 6. Plot hasil epoch ResNet50

Selanjutnya adalah evaluasi dari model yang di *training* dengan data *X_test dataset* gambar dan *y_test dataframe* genre anime, proses dan hail evaluasi pada *Gambar 7*.

```

1. _, acc = resnet_model.evaluate(X_test, y_test)
2. print("Accuracy = ", (acc * 100.0), "%")
3. loss, _ = resnet_model.evaluate(X_test, y_test)
4. print("Loss = ", (loss * 100.0), "%")
5.
6. 13/13 [=====] - 47s
4s/step - loss: 0.3662 - accuracy: 0.4802
7. Accuracy = 48.01980257034302 %
8. 13/13 [=====] - 47s
4s/step - loss: 0.3662 - accuracy: 0.4802
9. Loss = 36.62208318710327 %
10.
    
```

Gambar 7. Hasil evaluate ResNet50

Hasil dari *fungsi model.evaluate* dari kedua *data test*, didapatkan *Accuracy* sebesar 48% dan *Loss* sebesar 36.6%. model ini menunjukkan hasil yang cukup bagus dengan tingkat akurasi yang hampir mendekati 50%. Tahap terakhir adalah menyimpan model, untuk dipakai pada proses model *deployment* nanti.

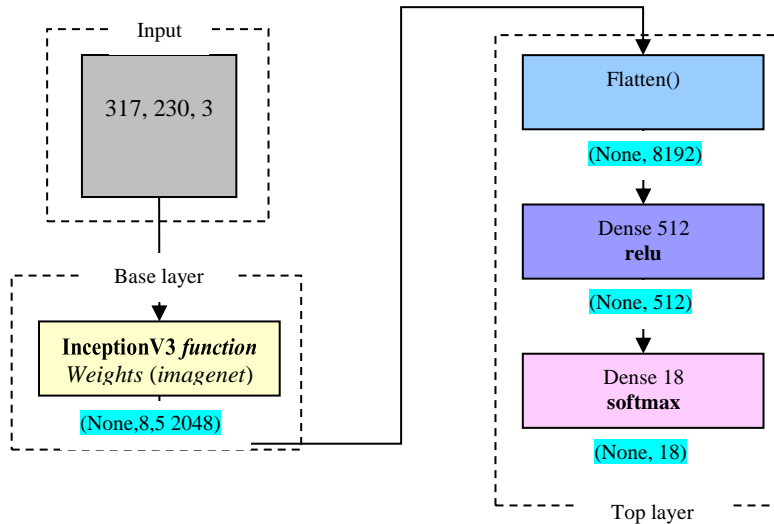
3.3. Model InceptionV3

Sama seperti sebelumnya pembangun model *neural network* diawali dengan *initialitation* model terlebih dahulu, dengan menggunakan fungsi *Sequential()*. Lalu menyiapkan *pre-trained ResNet50 model* dari *Keras API*.

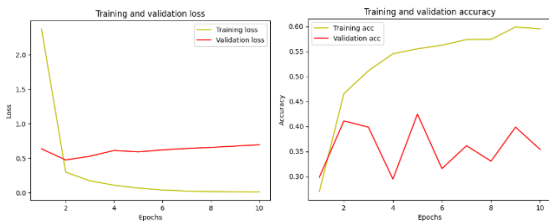
Penggunaan '*imagenet*' juga sama di implementasikan pada model ini, sebagai penambah bobot untuk model yang akan di bangun. *input_shape* ukurannya disamakan dengan *dataset* gambar. Lalu untuk *include_top* ini tidak akan dipakai, model ini akan menggunakan *dense layer* sendiri untuk menyesuaikan dengan banyaknya *classes*. jumlah

classes ini merupakan banyaknya genre yang di gunakan sebagai klasifikasinya. Pada model ini menggunakan *pooling default InceptionV3*. lalu *weights InceptionV3* ini akan di *disable* dan tidak akan melewati tahap Model Training, untuk mempersingkat waktu *training* karena terlalu banyaknya data, Gambar 8. Selain itu model ini

aktivasi *relu* dan *softmax* akan digunakan. Model ResNet50 ini menggunakan *Adam Optimizer*, loss menggunakan *binary_crossentropy*, dan *metrics* menggunakan perhitungan *accuracy*. Proses *training* yang dilakukan berjumlah 10 *epoch*. Dengan hasil pada Gambar 9.



Gambar 8. Flow model transfer learning InceptionV3



Gambar 9. Plot hasil epoch InceptionV3

Selanjutnya adalah tahap evaluasi dari model yang di *training* dengan data *X_test dataset* gambar dan *y_test dataframe* genre anime, proses dan hail evaluasi pada Gambar 10.

```

1. loss, acc = base_model.evaluate(X_test, y_test)
2. print("Accuracy = ", (acc * 100.0), "%")
3. print("Loss = ", (loss * 100.0), "%")
4.
5. 26/26 [=====] - 56s
2s/step - loss: 0.6920 - accuracy: 0.3540
6. Accuracy = 35.3960394859314 %
7. Loss = 69.20240521430969 %
8.
    
```

Gambar 10. Hasil evaluate InceptionV3

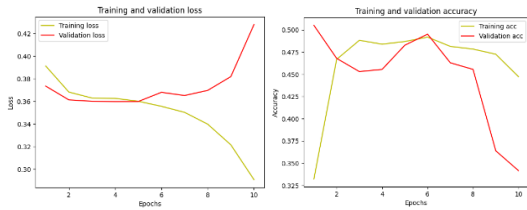
Hasil dari *function model.evaluate* dari kedua *data test* Gambar 9, didapatkan *Accuracy* sebesar 35% dan *Loss* sebesar 69%. Hasil dari *training* ini menunjukkan bahwa presentasi dari model ini kurang seimbang. Setelah evaluasi selesai selanjutnya adalah menyimpan model.

3.4. Model Custom

Custom Model dibuat dengan pertimbangan penggunaan dalam pengolahan gambar yang dipakai.

Serta sebagai perbandingan dengan model dari *Transfer Learning*. Pada layer pertama menggunakan *Convolutional Layers 2D* atau *Conv2D*. Dengan penggunaan filter 32 – 32 – 32 – 64 untuk 4 layer yang di pakai, filter ini berfungsi untuk mengenali yang akan di deteksi oleh model Gambar 12. *Pooling* yang digunakan adalah *MaxPooling* untuk mengurangi dimensi *input* dan jumlah parameter serta mempertahankan fitur-fitur penting dengan tujuan mencegah terjadinya *overfitting*. Lapisan *Flatten* juga digunakan untuk meratakan atau menyusun menjadi vektor. *Optimizer* yang dipakai adalah “adam”, Adam adalah hasil penurunan metode SGD yang didasarkan pada estimasi adaptif momen orde pertama dan kedua. algoritma optimasi yang menghitung tingkat pembelajaran secara adaptif untuk setiap parameter [25], dan *function* loss yang dipakai adalah “*binary_crossentropy*”. Jumlah *epoch* 10 dengan hasil pada Gambar 11 Layer yang mempunyai bobot matriks merupakan layer yang dapat di pelajari atau dapat di hitung. Total Parameter di sini merupakan total bobot yang di hasilkan oleh setiap layer. dalam *pooling layer* memang tidak menghasilkan parameter, namun bukan berarti layer ini tidak memiliki peran dalam *backpropagation*. Telah dijelaskan sebelumnya bahwa *pooling layer* ini bertanggung jawab untuk meneruskan nilai ke lapisan berikutnya dan sebelumnya selama *Feedforward* dan *backward propagation*.

Selanjutnya adalah tahap evaluasi dari model yang di *training* dengan data *X_test dataset* gambar dan *y_test dataframe* genre anime, proses dan hail evaluasi pada Gambar 13.

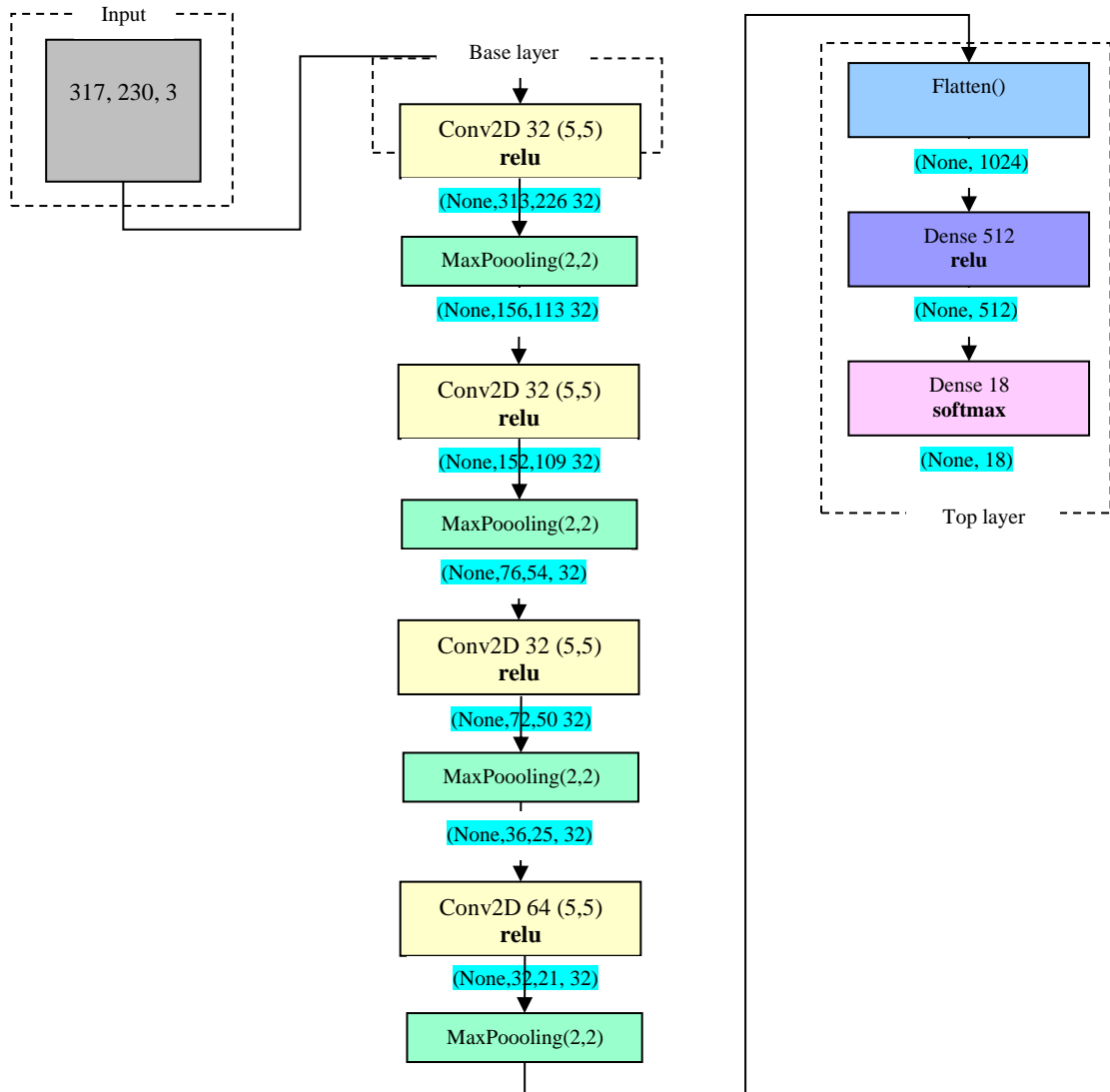


Gambar 11. Plot hasil *epoch* InceptionV3

```

1. loss, acc = base_model.evaluate(X_test, y_test)
2. print("Accuracy = ", (acc * 100.0), "%")
3. print("Loss = ", (loss * 100.0), "%")
4.
5. 13/13 [=====] - 6s
   465ms/step - loss: 0.4279 - accuracy: 0.3416
6. Accuracy = 34.15841460227966 %
7. Loss = 42.78874397277832 %
8.
    
```

Gambar 13. Hasil *evaluate Custom Model*



Gambar 12. *Flow model CNN Custom*

Hasil dari *function model.evaluate* dari kedua *data test*, didapatkan *Accuracy* sebesar 34% dan *Loss* sebesar 42%. Hasil dari *training* ini menunjukkan bahwa presentasi dari model ini tidak seimbang. Setelah evaluasi selesai selanjutnya adalah menyimpan model.

3.5. Perbandingan Evaluasi Model

Tahap ini merupakan tahap perbandingan dari setiap model yang sudah dibuat. Tingkat akurasi dan *loss* akan dibandingkan berdasarkan hasil yang didapat dari proses model evaluasi. *Function 'compile', 'fit', 'evaluate', dan 'save'* digunakan pada

tahap ini. *Function 'compile'* digunakan untuk menggabungkan arsitektur model yang sudah dibuat, *Function* ini membutuhkan 3 *hyperparameter* yang merpuakan perhitungan matematika, yaitu *Optimizer, loss* dan *metrics*. Dan *Function 'fit'* digunakan untuk *training*, ini juga membutuhkan *hyperparameter*, yaitu *epoch* yang berfungsi untuk menentukan berapa kali model akan membaca *dataset*, menghasilkan *accuracy* dan *loss*. *Accuracy* merupakan nilai dari hasil perhitungan akurasi yang didapat dari data yang melewati model. *Loss function* adalah fungsi untuk menggambarkan kekurangan terkait pada kemungkinan yang akan diproduksi oleh model. *Loss Function* dikatakan baik ketika fungsi menghasilkan

kesalahan sesuai dengan yang paling rendah [26]. Lalu Function 'evaluate' digunakan untuk melihat

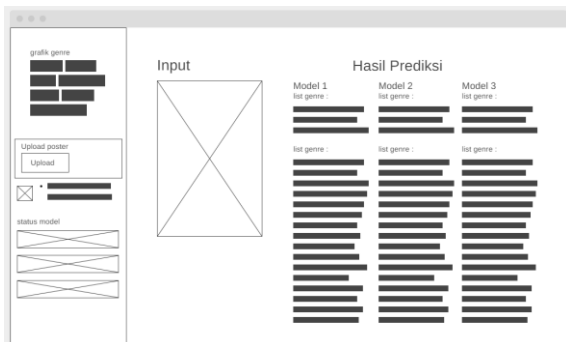
hasil *training dataset*, dan 'save' digunakan untuk menyimpan *file model* atau *weight model*.

Tabel 3. Perbandingan model

Model	optimizer	Loss funtion	Total epoch	accuracy	loss
Resnet50	adam	Binary crossentropy	10/10	0.4802	0.3662
InceptionV3	adam	Binary crossentropy	10/10	0.3540	0.6920
CustomNet	adam	Binary crossentropy	10/10	0.3415	0.4279

3.6. Deisgn UI

Aplikasi yang dibuat akan berbentuk *Landing Pagi* dan *design* menggunakan *wireframe* Gambar 14. *design* ini akan menjadi acuan pembuatan *website*. ini dibuat dengan tujuan memudahkan dalam pengembangan struktur dan fitur yang akan dibuat.



Gambar 14. Wireframe

Struktur tersebut meliputi semua *element HTML* yang akan digunakan. *HTML* adalah singkatan dari *HyperText Markup Language* yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman web, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah web Internet (*Browser*) [27]. Berikut ini adalah tabel penjelasan setiap *element* yang berisikan gambar dari *element*, jenis dan detail dari setiap *element* yang digunakan.

3.7. Implementasi Streamlit

Setelah model berhasil di buat dan di simpan sebagai *file*, maka tahap terakhir adalah menjadikan model ini sebagai aplikasi web. Web yang dibuat ini menggunakan bahasa pemrograman *python*, yaitu *framework streamlit*. Tahap pertama adalah membuat *function*, detail *code* pada Gambar 15. Ada dua jenis *function* yang akan dibuat pada aplikasi ini. *function* yang pertama adalah *function load_dataframe()*, ini merupakan *function* untuk memuat *dataset csv* anime. Dan yang terakhir adalah *function* untuk memuat *class* atau katagori dari genre anime, yaitu *function load_classes()*.

```

1. def load_dataframe():
2.     df = pd.read_csv('Datasets/anime.csv')
3.     df = df.sort_values(by=['Year'])
4.     df = df[df.Rating != 0]
5.     df = df[df.Studios != '[]']
6.     df = df[df.Genres != '[]']
7.     df = df[(df['Year'] >= 2013)]
8.     df["Image"] = df["Image"].str.replace("[?_&]", "", regex=True)
9.     return df
10.
11. def load_classes():
12.     class_df = pd.read_csv('Datasets/anime.csv')
13.     classes = class_df.drop(
14.         ['Image', 'Genres', 'Year',
15.          'Rating', 'Type', 'Format', 'Studios'], axis=1)
16.     return classes
17.

```

Gambar 15. Implementasi *streamlit* –pembuatan *function*

Lalu selanjutnya adalah pembuatan fitur *uploader*. Fitur *Upload File* ini dibuat menggunakan *st.file_uploader*, serta akan mendukung file gambar berjenis '*jpg*' dan '*png*'. Fitur ini berguna untuk mengunggah gambar poster yang akan di prediksi genrenya.

```

1. file = st.file_uploader("Upload Poster :", type=["jpg", "png"])
2.

```

Gambar 16 Implementasi *streamlit* –file uploader

Dan terakhir adalah fitur prediksi yang cukup sederhana. Yaitu dengan memanggil *function load_dataframe()* dan *function load_classes()* Gambar 17. Selanjutnya *file gambar inputan* dari fitur *Upload File* akan di ubah menjadi data *array* tiga dimensi dan akan di lakukan pembagian dengan 255. Lalu *list genre* akan di ambil dari nama pada *column dataset anime*.

```

1. # MAIN
2. df = load_dataframe()
3. list_class = load_classes()
4.
5. # input
6. input_image = Image.open(file)
7. imgori = image.load_img(file, target_size=(height, width, 3))
8. img = image.img_to_array(imgori)
9. img = img/255.
10. img = np.expand_dims(img, axis=0)
11.
12. # init class
13. classes = np.array(list_class.columns)
14.

```

Gambar 17 image file to array

Lalu gambar yang berupa *array 3 dimensi* tersebut akan melalui tahap *model.predict*, ini akan memadatkan gambar dan menyisakan informasi dari gambar tersebut dalam berbentuk *array biasa*, detail *code* pada Gambar 18.

```

1. proba1 = model1.predict(img)
2. sorted_categories1 = np.argsort(proba1[0])[::-1]
3.

```

Gambar 18 Model predict

Setiap angka yang ada pada *array* itu akan menjadi perwakilan untuk setiap genre yang ada pada *function load_classes*. Hasil akhirnya angka itu di persentasekan dan dihitung *Gambar 19*, dan ditampilkan pada *Landing Page*, *Gambar 20*.

```

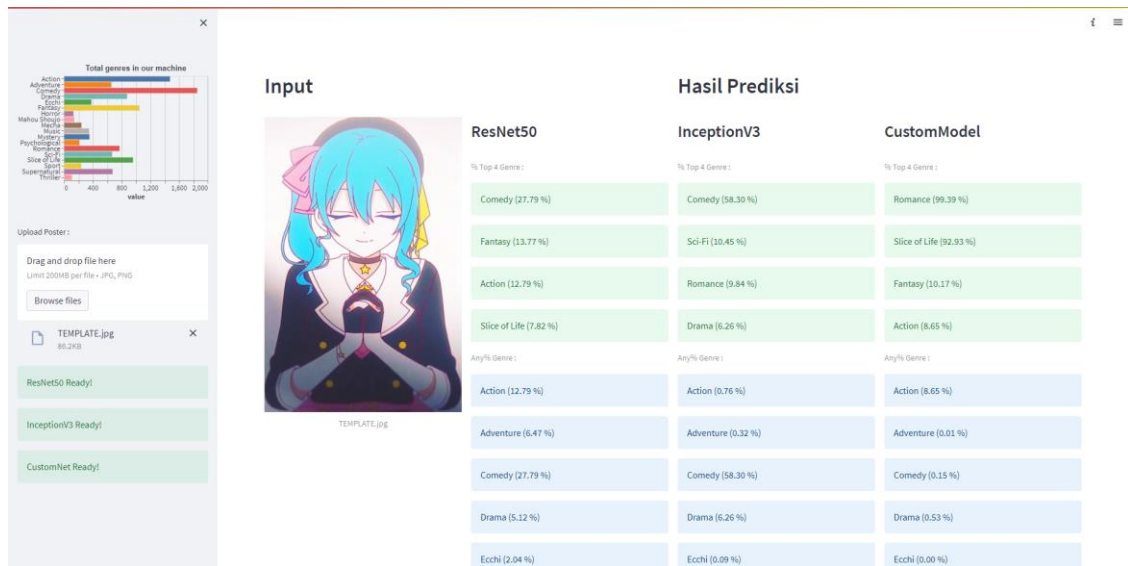
1. genre1 = []
2. for i in range(4):
3.     st.success("{}".format(classes[sorted_categories1[i]]+"
({:.2f} %)".format(probal[0][sorted_categories1[i]] * 100))
4. for i in range(18):
5.     list1 = ("{}".format(classes[sorted_categories1[i]]+"
({:.2f} %)".format(probal[0][sorted_categories1[i]] * 100))
6.     genre1.append(list1)
7. genre1 = sorted(genre1)
8. st.caption("Any% Genre : ")
9. for x in range(len(genre1)):
10.    st.info(genre1[x])
11.
    
```

Gambar 19 Kalkulasi Hasil

3.8. Hasil Implementasi

Hasil dari penelitian ini berupa *Landig page* yang merupakan tampilan dari aplikasi yang

dibangun berdasarkan *wireframe Gambar 18*. Pada *Landing page* terdapat *Sidebar* dan halaman utama untuk menampilkan gambar yang di *upload* dan persentase prediksi genrenya. Lalu pada bagian samping terdapat *Sidebar* yang memuat fitur utama dari aplikasi ini. yaitu fitur *Upload Poster*, fitur ini berfungsi untuk mengunggah gambar poster yang akan di prediksi. Terakhir terdapat grafik dari total genre yang ada pada *dataset*. Halaman *Landing Page* akan berubah jika telah mendapat masukan dari fitur *upload*. pada halaman utama akan tampil gambar yang telah di *upload* dan akan tampil *list* berupa genre-genre yang telah di prediksi. *List* genre dibagi menjadi dua bagian, *list* pertama merupakan *list* genre yang mempunyai persentase tertinggi, dan *list* genre kedua merupakan *list* yang memuat presentasi dari masing-masing genre yang ada.






Gambar 20. halaman *Landing Page* dengan hasil

Pengujian model dilakukan dengan tujuan untuk mengetahui performa dari masing-masing model *Resnet50*, *InceptionV3* dan *CustomModel*. Data Poster mengambil sampel poster acak dari *website anilist.co*, sampel poster yang diambil adalah poster anime yang rilis pada tahun 2022 agar tidak sama dengan data *test* sebelumnya. Pengujian ini dilakukan

menggunakan aplikasi yang telah *running* di *localhost*, dengan cara *input gambar*. Genre yang berhasil terprediksi akan dipilih dari 4 nilai prediksi tertinggi. seperti pada *Gambar 18*. *List* genre yang berhasil terprediksi tersebut akan ditampilkan pada kolom *Prediksi Genre* di *Tabel 4*.

Tabel 4. Data uji poster

No	Poster	Judul	Genre	Prediksi Genre		
				Resnet50	InceptionV3	CustomModel
1.		<i>Mamahaha no Tsurego ga Motokano datta</i>	- Comedy - Romance	Comedy (25.80 %) Action (18.38 %) Fantasy (10.60 %) Sci-Fi (7.02 %)	Comedy (81.75 %) Romance (74.15 %) % Slice of Life (35.67 %) % Sci-Fi (31.16 %)	Romance (98.17 %) Ecchi (73.97 %) Comedy (58.65 %) Drama (5.73 %)

2.		<i>Cyberpunk: Edgerunners</i>	- Action - Drama - Psychological - Sci-Fi	Comedy (31.17 %) Action (19.11 %) Fantasy (10.24 %) Sci-Fi (6.24 %)	Adventure (24.04 %) Action (11.19 %) Ecchi (5.34 %) Drama (3.62 %)	Sci-Fi (99.91 %) Drama (89.39 %) Supernatural (9.85 %) Sports (1.25 %)
3.		<i>Chainsaw Man</i>	- Action - Comedy - Drama - Horror - Supernatural	Action (30.77 %) Fantasy (12.68 %) Comedy (9.50 %) Sci-Fi (8.14 %)	Action (99.29 %) Drama (93.29 %) Psychological (90.62 %) Mystery (85.10 %)	Action (99.91 %) Sci-Fi (99.65 %) Adventure (78.37 %) Drama (42.46 %)
4.		<i>ONE PIECE FILM: RED</i>	- Action - Adventure - Comedy - Drama - Fantasy - Music	Action (46.15 %) Fantasy (16.91 %) Adventure (8.40 %) Sci-Fi (7.27 %)	Comedy (98.49 %) Drama (25.20 %) Supernatural (18.42 %) Music (3.35 %)	Sci-Fi (99.96 %) Drama (99.89 %) Comedy (93.20 %) Action (92.31 %)
5.		<i>Bocchi the Rock!</i>	- Comedy - Music - Slice of Life	Comedy (25.71 %) Fantasy (16.49 %) Action (15.38 %) Adventure (7.95 %)	Supernatural (93.33 %) Music (49.20 %) Mystery (24.85 %) Sci-Fi (12.47 %)	Comedy (99.87 %) Romance (26.23 %) Supernatural (15.53 %) Action (4.17 %)

Tabel 4 menunjukkan, masing-masing model berhasil memprediksi genre yang ada, namun tidak semua genre dari anime berhasil diprediksi. CustomModel mempunyai performa yang baik dengan memberikan persentase tertinggi pada setiap genre yang diprediksi. InceptionV3 mempunyai performa yang bagus, terutama pada kemampuannya dalam memprediksi genre Music, yang tidak dapat di prediksi oleh CustomModel dan ResNet50. ResNet50 sendiri tidak terlalu buruk, karena berhasil memprediksi genre anime yang ada, meskipun dengan nilai persentase yang kecil.

4. DISKUSI

Berdasarkan hasil penelitian, metode Transfer Learning ResNet50 dan InceptionV3 memiliki hasil yang baik jika dibandingkan dengan arsitektur CNN Custom Model yang telah dibuat. Jika mengacu pada yang di kemukakan oleh Septiana Rizky tentang dataset dimana Jumlah class tidak seimbang merupakan suatu kondisi di mana terdapat dataset yang jumlah kelasnya terdapat perbedaan yang signifikan terhadap masing-masing jumlah kelas [28], berakibat pada kinerja model yang kurang maksimal karena keterbatasan dalam limitasi data, terutama data genre yang menjadi klasifikasi dan class utama.

Artinya adalah keterbatasan dataset akan mempengaruhi tingkat akurasi, dampak dari jumlah dari klasifikasi yang tidak seimbang menjadi sumber terjadinya penurunan pada kinerja model CNN [29].

5. KESIMPULAN

Penggunaan Metode Transfer Learning dapat di implementasikan untuk memprediksi genre anime, dengan menggunakan model atau arsitektur ResNet50, InceptionV3 dan custom model. Top Layer pada kedua arsitektur tersebut untuk menyesuaikan dengan banyaknya klasifikasi genre. ResNet50 merupakan arsitektur yang memiliki kinerja yang cukup baik berdasarkan hasil evaluasi model, dengan nilai accuracy 0.4802 dan loss 0.3662. sedangkan InceptionV3 berdasarkan hasil evaluasi model, memiliki kinerja yang kurang dengan nilai accuracy 0.3540 dan loss 0.6920. Sedangkan Model Custom memiliki tingkat accuracy 0.3415 dan loss 0.4279. ketiga model juga berhasil memprediksi genre dengan hasil yang beragam. Dan untuk framework streamlit sendiri dapat di gunakan untuk memuat model neural network yang dibuat. Dengan penggunaan fitur st.file_uploader gambar dapat diunggah dan di baca oleh model yang sudah di load, sehingga proses prediksi dapat dilakukan. Hasil

predikis dapat ditampilkan berbentuk *list column* menggunakan fitur *st.columns*.

DAFTAR PUSTAKA

- [1] D. F. N. Pratama and A. K. Adim, "Konsep Diri Mahasiswa Otaku Di Kota Bandung (Analisis Terhadap Konsep Diri Yang Dimiliki Oleh Mahasiswa Otaku Yang Ada Dikota Bandung)," Vol. 13, 2022.
- [2] I. Fansuri, "Penerapan Algoritma Smith Pada Aplikasi Pencarian Anime Dalam Kumpulan Anime," *Terapan Informatika Nusantara*, Vol. 1, No. 6, Pp. 333–338, 2020, [Online]. Available: <https://ejournal.seminarid.com/index.php/tin>
- [3] A. M. Ashari And I. Rochmawati, "Analisis Visual Pada Poster Promosi Film Trilogi Batman," 2022. [Online]. Available: <https://ojs.unikom.ac.id/index.php/divagatra>
- [4] D. Kurniawan, S. F. Seni, D. Desain, S. Desain, And K. Visual, "STUDI SEMIOTIKA KARAKTER MONOKUMA PADA ANIME FRANCHISE SERIES 'DANGANRONPA,'" 2019. [Online]. Available: <http://publikasi.dinus.ac.id/index.php/andharupa>
- [5] D. Alita, Y. Rahmanto, And A. Dwi Putra, "PELATIHAN DESAIN GRAFIS PADA SMKS NURUL HUDA PRINGSEWU," *Journal Of Technology And Social For Community Service (JTSCS)*, Vol. 3, No. 2, Pp. 337–346, 2022, [Online]. Available: <https://ejournal.teknokrat.ac.id/index.php/teknobdimas>
- [6] R. S. Hartono And H. Armanto, "Klasifikasi Genre Manga Menggunakan Convolutional Neural Network," Vol. 8, No. 2, Pp. 573–584, 2021, [Online]. Available: <http://jurnal.mdp.ac.id>
- [7] D. Sebastian, "Implementasi Algoritma K-Nearest Neighbor Untuk Melakukan Klasifikasi Produk Dari Beberapa E-Marketplace," Vol. 5, Pp. 2443–2229, 2019, Doi: 10.28932/jutisi.v5i1.913.
- [8] D. M. Wonohadidjojo, "Perbandingan Convolutional Neural Network Pada Transfer Learning Method Untuk Mengklasifikasikan Sel Darah Putih," *Ultimatics : Jurnal Teknik Informatika*, Vol. 13, No. 1, P. 51, 2021.
- [9] Sunario Megawan And Wulan Sri Lestari, "Deteksi Spoofing Wajah Menggunakan Faster R-CNN Dengan Arsitektur Resnet50 Pada Video," *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, Vol. 9, No. 3, Pp. 261–267, Aug. 2020, Doi: 10.22146/v9i3.231.
- [10] F. Royana, P. Yuniar Maulida, R. Nurul Hasanah, And S. Setia Rahayu, "Aplikasi Mobile Deteksi Dini Kanker Kulit Berdasarkan Image Processing | 100," 2021. [Online]. Available: <http://journal.pwmjateng.com/index.php/jle>
- [11] A. Michael, "Journal Dynamic Saint Komparasi Kombinasi Pre-Trained Model Dengan SVM Pada Klasifikasi Kematangan Kopi Berbasis Citra," Vol. 7, No. 1, 2022, Doi: 10.47178/dynamicsaint.v5xx.xxxx.
- [12] S. Ramaneswaran, K. Srinivasan, P. M. D. R. Vincent, And C. Y. Chang, "Hybrid Inception V3 Xgboost Model For Acute Lymphoblastic Leukemia Classification," *Computational And Mathematical Methods In Medicine*, Vol. 2021. Hindawi Limited, 2021. Doi: 10.1155/2021/2577375.
- [13] C. W. T. Koh, J. S. G. Ooi, G. L. C. Joly, And K. R. Chan, "Gene Updater: A Web Tool That Autocorrects And Updates For Excel Misidentified Gene Names," *Sci Rep*, Vol. 12, No. 1, Dec. 2022, Doi: 10.1038/s41598-022-17104-3.
- [14] A. Aboah, M. Boeding, And Y. Adu-Gyamfi, "Mobile Sensing For Multipurpose Applications In Transportation," 2021.
- [15] S. Lestari And S. Saepudin, "ANALISIS SENTIMEN VAKSIN SINOVAQ PADA TWITTER MENGGUNAKAN ALGORITMA NAIVE BAYES," 2021. [Online]. Available: <https://vaksin.kemkes.go.id/>
- [16] A. Peryanto, A. Yudhana, And R. Umar, "Klasifikasi Citra Menggunakan Convolutional Neural Network Dan K Fold Cross Validation," 2020. [Online]. Available: <http://jurnal.polibatam.ac.id/index.php/jaic>
- [17] N. D. Miranda, L. Novamizanti, And S. Rizal, "CONVOLUTIONAL NEURAL NETWORK PADA KLASIFIKASI SIDIK JARI MENGGUNAKAN RESNET-50," *Jurnal Teknik Informatika (Jutif)*, Vol. 1, No. 2, Pp. 61–68, Dec. 2020, Doi: 10.20884/1.jutif.2020.1.2.18.
- [18] F. Nurona Cahya *Et Al.*, "SISTEMASI: Jurnal Sistem Informasi Klasifikasi Penyakit Mata Menggunakan Convolutional Neural Network (CNN)," 2021. [Online]. Available: <http://sistemasi.ftik.unisi.ac.id>
- [19] Rizki Putra Pamungkas, Dwina Kuswardani, And Riki Ruli A Siregar, "Penerapan Algoritma Backpropagation Pada Pengenalan Tanda Nomor Kendaraan Bermotor Untuk Kartu Parkir Berbasis RFID," 2020.
- [20] M. Hasbi Ashshiddieqy And A. Rizal,

- “Klasifikasi Suara Paru Dengan Convolutional Neural Network (CNN),” 2020.
- [21] M. V. Al Hasri And E. Sudarmilah, “Sistem Informasi Pelayanan Administrasi Kependudukan Berbasis Website Kelurahan Banaran,” *MATRIK: Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer*, Vol. 20, No. 2, Pp. 249–260, May 2021, Doi: 10.30812/Matrik.V20i2.1056.
- [22] M. Agarina And A. Suryadi Karim, “Seminar Nasional Hasil Penelitian Dan Pengabdian 2019 IBI DARMAJAYA Bandar Lampung,” 2019.
- [23] R. Pratama, A. Fuad Assagaf, And F. Tempola, “DETEKSI KEMATANGAN BUAH TOMAT BERDASARKAN FITUR WARNA MENGGUNAKAN METODE TRANSFORMASI RUANG WARNA HIS,” *Jurnal Informatika Dan Komputer) P-ISSN*, Vol. 2, No. 2, Pp. 2355–7699, 2019, Doi: 10.33387/Jiko.
- [24] D. Agusti And A. A. Nababan, “Penerapan Metode Harmonic Mean Filter Dalam Mereduksi Gaussian Noise Pada Citra Digital,” *Jurnal Nasional Komputasi Dan Teknologi Informasi*, Vol. 5, No. 3, 2022.
- [25] N. D. Miranda, L. Novamizanti, And S. Rizal, “CONVOLUTIONAL NEURAL NETWORK PADA KLASIFIKASI SIDIK JARI MENGGUNAKAN RESNET-50,” *Jurnal Teknik Informatika (Jutif)*, Vol. 1, No. 2, Pp. 61–68, Dec. 2020, Doi: 10.20884/1.Jutif.2020.1.2.18.
- [26] B. Yanto *Et AL.*, “Klasifikasi Tekstur Kematangan Buah Jeruk Manis Berdasarkan Tingkat Kecerahan Warna Dengan Metode Deep Learning Convolutional Neural Network,” Vol. 6, No. 2, P. 2021, 2021.
- [27] C. M. Lengkong, R. Sengkey, And A. Sugiarto, “Sistem Informasi Pariwisata Berbasis Web Di Kabupaten Minahasa,” *Jurnal Teknik Informatika*, Vol. 14, No. 1, 2019.
- [28] P. Septiana Rizky, R. Haiban Hirzi, U. Hidayaturrohmah, U. Hamzanwadi Selong Jl TGKH Muhammad Zainuddin Abdul Madjid Pancor, And L. Timur, “Perbandingan Metode Lightgbm Dan Xgboost Dalam Menangani Data Dengan Kelas Tidak Seimbang,” 2022. [Online]. Available: www.unipasby.ac.id
- [29] K. De Angeli *Et AL.*, “Class Imbalance In Out-Of-Distribution Datasets: Improving The Robustness Of The Textcnn For The Classification Of Rare Cancer Types,” *J Biomed Inform*, Vol. 125, Jan. 2022, Doi: 10.1016/J.Jbi.2021.103957.