

Implementasi *Convolutional Neural Network* untuk Klasifikasi Pembalap MotoGP Berbasis GPU

Afandi Nur Aziz Thohari¹, Galuh Boy Hertantyo²

^{1,2}S1 Rekayasa Perangkat Lunak, Fakultas Teknologi Industri dan Informatika,
Institut Teknologi Telkom Purwokerto

^{1,2}JL. D.I Panjaitan No. 128 Purwokerto 53147

afandi@ittelkom-pwt.ac.id, galuh@ittelkom-pwt.ac.id

Abstrak – Salah satu bidang keilmuan baru dalam *machine learning* yang berkembang akibat perkembangan teknologi GPU adalah *deep learning*. Adanya *deep learning* dapat menyelesaikan salah satu masalah dalam visi komputer yaitu klasifikasi objek pada citra. Klasifikasi citra sendiri dapat dikatakan merupakan pekerjaan yang sangat sulit dilakukan komputer karena pada dasarnya komputer hanya menangkap data citra sebagai rentetan nilai dan piksel. Oleh karena itu pada penelitian ini diimplementasikan metode *Convolutional Neural Network* (CNN) sebagai salah satu model dari *deep learning* yang memiliki kemampuan yang baik dalam klasifikasi data dengan struktur dua dimensi seperti citra. CNN sendiri pada dasarnya merupakan arsitektur jaringan syaraf tiruan yang memerlukan proses pelatihan berulang-ulang untuk mendapatkan akurasi yang baik. Proses pelatihan dengan CNN akan membutuhkan komputasi yang sangat berat serta waktu yang lama, untuk itu penggunaan performa GPU sangatlah dibutuhkan untuk mempercepat waktu pelatihan. Berdasarkan hasil pengujian menggunakan *epoch* sejumlah 150 terhadap citra pembalap MotoGP, menunjukkan akurasi sebesar 96,67% terhadap data test dengan waktu pelatihan mencapai 473 detik. Sehingga dapat disimpulkan bahwa teknik *Deep Learning* dengan CNN mampu melakukan klasifikasi citra pembalap MotoGP dengan sangat baik. Sedangkan penggunaan GPU dapat mempercepat proses komputasi hingga 11 kali lipat dibanding jika menggunakan CPU.

Kata kunci – citra, CNN, *deep learning*, GPU, klasifikasi

Abstract – One of the new science field in machine learning which evolve because of GPU technology is deep learning. The existence of deep learning can resolve one of problem in computer vision that is object classification in image. Image Classification can be said to be very difficult job that done by computer because basically computer only capture image data as a series of values and pixels. Therefore in this research, the *Convolutional Neural Network* (CNN) method is implemented as a model of deep learning that has good ability in data classification with two-dimensional structures such as image. CNN is basically an artificial neural network architecture that requires repeated training processes to get good accuracy. The training process with CNN will require very heavy computation and a long time consume, therefore the use of GPU performance is needed to speed up training time. Based on the results of testing using 150 epochs on the image of MotoGP racers, it shows an accuracy of 96.67% of the test data with training time reaching 473 seconds. So it can be concluded that the Deep Learning technique with CNN is able to classify the image of MotoGP drivers very well. While the use of GPU can speed up the computation process up to 11 times compared to using a CPU.

Keywords – Classification, image, CPU, GPU.

I. PENDAHULUAN

Pada saat ini tidak dapat dipungkiri bahwa perkembangan teknologi informasi sangat cepat. Salah satu contoh perkembangan tersebut adalah pemanfaatan GPU yang dapat digunakan untuk melakukan pemrosesan paralel. GPU yang biasa digunakan untuk bermain *game*, ternyata dapat pula digunakan untuk melakukan pemrosesan teks, citra, video dan lain sebagainya. Perkembangan teknologi GPU lambat laun semakin meningkat. Hal ini dibuktikan dengan terus bertambahnya *core* dalam GPU yang membuat waktu pemrosesan semakin cepat. Hal ini tentu dimanfaatkan oleh para peneliti di bidang *computer vision* untuk menurunkan waktu training yang biasanya lebih lama jika diproses menggunakan CPU.

Selain perkembangan *hardware* dalam meningkatkan performa komputer, terdapat pula perkembangan *software* yang mampu meniru kecerdasan manusia (kecerdasan buatan). Salah satu bentuk kecerdasan komputer tersebut adalah mampu untuk mengklasifikasi citra yang diberikan. Klasifikasi citra merupakan salah satu bagian permasalahan dalam *computer vision*. Ide dari klasifikasi citra sendiri adalah dengan memberi komputer masukan berupa citra dan akan merepresentasikan kategori dari citra tersebut. Terlintas hal ini terlihat seperti pekerjaan yang sepele dan mudah bagi manusia, namun kenyataannya hal ini merupakan pekerjaan yang sangat sulit bagi komputer. Permasalahan ini dikarenakan komputer hanya melihat citra berupa deretan nilai dan data piksel.

Salah satu solusi untuk mengatasi masalah tersebut adalah dengan menggunakan *machine learning* sebagai salah satu teknik pembelajaran. *Machine learning* merupakan cabang dari kecerdasan buatan yang fokus pada pembelajaran sendiri (*self learning*) atau tanpa harus berulang kali diprogram oleh manusia. Akan tetapi, hanya menggunakan *machine learning* saja masih belum cukup untuk mengatasi klasifikasi citra. Hal ini dikarenakan data yang diperlukan untuk mengenal sebuah citra begitu kompleks dan banyak. Variasi dari citra yang ingin dilatih memiliki perbedaan bentuk dan pola sehingga sulit mendapatkan hasil yang baik. Selain itu kesamaan pola dan bentuk juga dapat menyebabkan kesalahan dalam prediksi dikarenakan fitur yang dipelajari membuat model tidak dapat melakukan prediksi dengan baik.

Model pembelajaran untuk kasus tersebut akan lebih baik apabila menerapkan konsep jaringan syaraf tiruan dengan banyak lapisan. Maka disinilah penggunaan teknik *deep learning* diperlukan. Pada dasarnya *deep learning* adalah implementasi konsep dasar *machine learning* yang mengadaptasikan algoritma jaringan syaraf tiruan dengan lapisan yang lebih banyak. Perbedaananya, *deep learning* memakai lebih banyak lapisan tersembunyi yang digunakan

antara lapisan masukan dan lapisan keluaran, oleh karena itu jaringan tersebut dinamakan *deep neural net*.

Salah satu metode *deep learning* yang dapat digunakan untuk melakukan klasifikasi citra adalah *Convolutional Neural Network* (CNN). Saat ini metode CNN memiliki hasil paling akurat dalam pengenalan citra. Hal tersebut dikarenakan CNN berusaha meniru sistem pengenalan citra pada visual cortex manusia, sehingga memiliki kemampuan mengolah informasi citra [1]. Adapun salah satu penelitian mengenai pengolahan citra dengan metode CNN yang mendapatkan hasil akurasi bagus yaitu penelitian yang dilakukan oleh [2] untuk klasifikasi citra candi yang menghasilkan akurasi citra uji sebesar 85,57%. Kemudian penelitian terbaru dilakukan oleh [3] yang melakukan identifikasi citra wayang punakawan dan menghasilkan akurasi citra uji sebesar 91,6%.

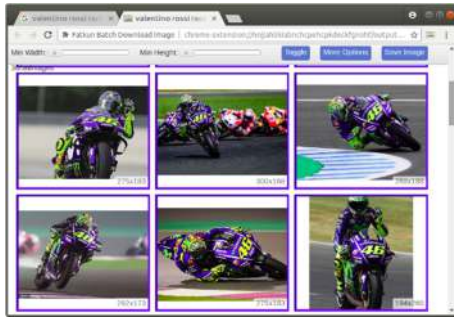
CNN memiliki jaringan khusus yang dinamakan jaringan konvolusi, dimana pada lapisan ini citra masukan diolah berdasarkan *filter* yang sudah ditentukan. Dari setiap lapisan ini akan menghasilkan sebuah pola dari beberapa bagian citra yang nantinya akan lebih mudah untuk diklasifikasi. Teknik ini dapat membuat fungsi pembelajaran citra menjadi lebih efisien. Berdasarkan kelebihan tersebut, maka pada penelitian ini digunakan metode CNN sebagai solusi untuk klasifikasi pembalap MotoGP. Arsitektur CNN yang digunakan pada penelitian ini adalah varian jaringan VGGNet yang lebih kecil dan ringkas [4]. Agar mempercepat waktu pelatihan dan supaya perangkat keras dari komputer lebih awet, maka digunakan GPU sebagai tempat pemrosesannya.

II. METODE PENELITIAN

Metodologi pada penelitian ini diklasifikasikan dalam empat kriteria yaitu pengumpulan data, proses pengolahan data, analisa data, dan evaluasi. Adapun penjelasan dari masing-masing kriteria adalah sebagai berikut.

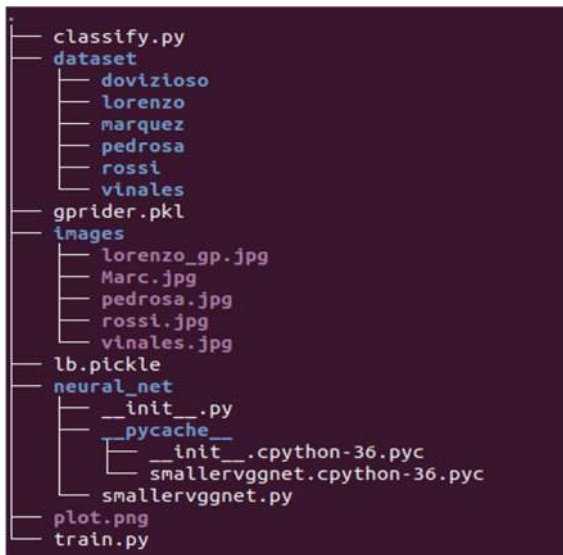
A. Pengumpulan Data

Pengumpulan data dilakukan dengan menggunakan *software fatkun batch image downloader* yang merupakan plugin di *browser google chrome*. Data yang diambil adalah data pembalap-pembalap MotoGP yang terbagi menjadi enam kategori yaitu rossi, lorenzo, dovizioso, marquez, pedrosa dan vinales. Jumlah sample pada setiap kategorinya adalah sebanyak 50 data citra. Menggunakan *google image* akan lebih memudahkan peneliti mendapatkan citra yang ada pada google, selanjutnya gambar yang telah dicari pada *google image* akan diunduh dengan cepat menggunakan *fatkun batch image downloader*. Gambar 1 merupakan tampilan data citra yang siap diunduh.

Gambar 1. Data citra pada *fatkun batch image downloader*

B. Metode Analisis Data

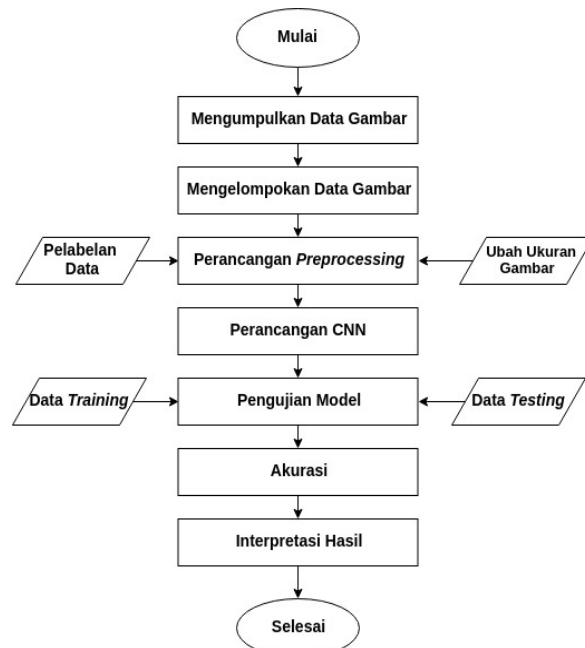
Metode analisis yang digunakan dalam penelitian ini adalah metode *deep learning* yaitu menggunakan algoritma *convolutional neural network* (CNN). Analisis dilakukan ketika data berupa gambar pembalap dipisahkan kedalam direktori bernama dataset. Dimana didalam dataset gambar pembalap dikelompokkan lagi berdasarkan nama pembalap. Gambar 2 menunjukkan struktur dari proyek.



Gambar 2. Struktur direktori dari proyek

C. Tahapan Penelitian

Tahapan atau langkah-langkah dalam penelitian ini ditunjukkan pada Gambar 3.



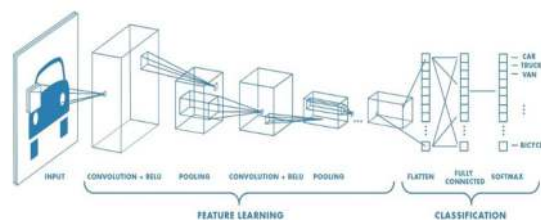
Gambar 3. Tahapan Penelitian

Arsitektur CNN yang digunakan pada penelitian ini adalah varian jaringan *Verry Deep Convolutional Network* (VGGNet) yang lebih kecil dan ringkas. Adapun arsitektur VGGNet memiliki karakteristik sebagai berikut.

1. Hanya menggunakan *3x3 convolutional layer*
2. Mengurangi *volume size* dengan *max pooling*
3. Menggunakan *fully connected layer* pada akhir jaringan sebelum klasifikasi softmax

D. Rancangan Convolutional Neural Network (CNN)

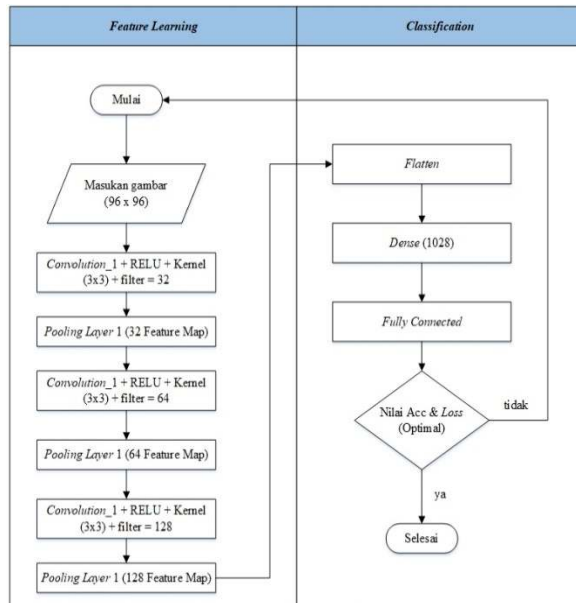
Setelah dilakukan analisa data langkah selanjutnya adalah melakukan pelatihan model CNN. Umumnya dalam CNN memiliki 2 tahapan yaitu tahap *feature learning* dan *classification*. Pada penelitian ini masukan gambar pada model CNN menggunakan citra yang berukuran $96 \times 96 \times 3$. Angka tiga yang dimaksud adalah sebuah citra yang memiliki 3 *channel* yaitu *Red*, *Green*, dan *Blue* (RGB). Citra masukan kemudian akan diproses terlebih dahulu melalui proses konvolusi dan proses pooling pada tahap *feature learning*. Lebih jelas mengenai arsitektur CNN dapat dilihat pada Gambar 4.



Gambar 4. Arsitektur Klasifikasi Gambar

Jumlah proses konvolusi pada rancangan ini memiliki dua lapisan konvolusi. Setiap konvolusi memiliki jumlah filter dan ukuran kernel yang

berbeda. Kemudian dilakukan proses *flatten* atau proses mengubah *feature map* hasil *pooling layer* kedalam bentuk vektor. Proses ini biasa disebut dengan tahap *fully connected layer*. Gambar 5 menunjukkan rancangan dari arsitektur CNN pada penelitian ini.



Gambar 5. Rancangan Arsitektur CNN

E. Perangkat Keras dan Lunak

Proses klasifikasi citra untuk memprediksi pembalap MotoGP membutuhkan spesifikasi perangkat lunak dan perangkat keras tertentu. Utamanya pada perangkat keras, membutuhkan tempat memproses data berkecepatan tinggi. Lebih jelas mengenai spesifikasi perangkat lunak dan keras yang digunakan pada penelitian ini, ditunjukkan pada Tabel 1

Tabel 1. Spesifikasi Perangkat Lunak dan Keras

Perangkat Lunak	Python 3.6.5
	Open CV 3.4.1
	Tensorflow-gpu 1.8.0
	Keras 2.2.0
Perangkat Keras	Intel Core i3-5005U CPU
	GeForce 930M 2 Gb
	12 b RAM

III. HASIL PENELITIAN

A. Pengaruh Jumlah Epoch

Epoch adalah ketika seluruh *dataset* sudah melalui proses *training* pada *Neural Network* sampai dikembalikan ke awal dalam satu putaran. Dalam *Neural Network* satu *epoch* itu terlalu besar dalam proses pelatihan karena seluruh data diikuti kedalam proses *training* sehingga akan membutuhkan

waktu cukup lama. Agar mempermudah dan mempercepat proses *training*, biasanya data rate dibagi per *batch* (*Batch Size*). Penentuan nilai dari *batch size* biasanya tergantung peneliti dengan melihat banyak sampel.

Tabel 2. Perbandingan Akurasi dan Waktu pada Berbagai Jumlah Iterasi (*Epoch*)

Epoch	Akurasi	Waktu (s)	
		CPU	GPU
50	79,8%	2736	201
75	84,2%	3545	305
100	88,8%	4646	405
125	85,7%	6674	492
150	90,5%	6635	473

Pelatihan citra menggunakan *epoch* biasanya memakan waktu yang sangat lama apabila menggunakan CPU biasa. Hal itu dapat dilihat pada Tabel 2 yang menunjukkan perbandingan waktu pelatihan antara CPU dan GPU dengan jumlah *epoch* yang bervariasi. Penggunaan GPU dapat mempercepat proses *training* sampai 11 kali lipat dibanding jika menggunakan CPU. Banyaknya jumlah *epoch* ternyata juga mempengaruhi akurasi dari identifikasi citra yang diuji. Semakin banyak jumlah *epoch*, maka akan semakin akurat pula identifikasi citra yang ditampilkan. Namun terkadang jumlah *epoch* besar tidak menjamin akurasi dapat juga besar, hal itu dipengaruhi oleh banyaknya *dataset*.

B. Pengaruh Learning Rate

Penentuan *learning rate* ini sangat berpengaruh pada performa akurasi. Berdasarkan table 3 ,penggunaan nilai *learning rate* 0.01 menghasilkan nilai akurasi yang kurang optimal yaitu sebesar 80,6%. Hal ini dikarenakan semakin besar *learning rate*, maka akan menghasilkan nilai *loss* yang besar ketika menjalankan iterasi pada saat *training*. Sebaliknya, ketika menggunakan nilai *learning rate* 0.001 tingkat akurasi yang dihasilkan besar yaitu 88,8%. Hal ini disebabkan adanya beberapa nilai *loss* yang mulai menurun dalam beberapa iterasi.

Tabel 3. Perbandingan Learning Rate

Learning Rate	Nilai Akurasi	Waktu (s)	
		CPU	GPU
0.01	80.6%	4986	416
0.001	88.8%	5309	405
0.0001	74.4%	5403	515

Berbeda dengan penggunaan *learning rate* 0.0001 yang menunjukkan tingkat akurasi yang kurang optimal yaitu sebesar 74,4%. Hal ini tentunya disebabkan oleh lambatnya proses konvergensi nilai *loss* pada saat proses *training*. Kemudian dari segi waktu, semakin kecil *learning rate* maka semakin lama waktu yang digunakan untuk *training*.

C. Akurasi Model Data Testing

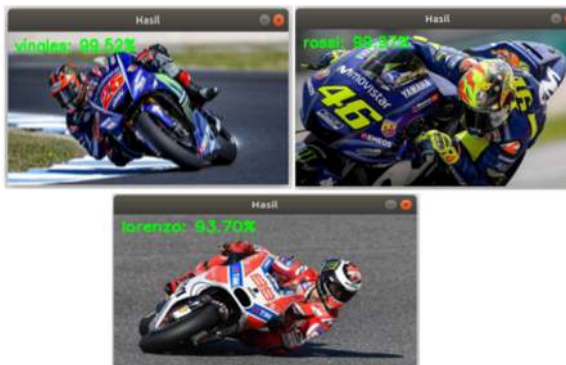
Data yang digunakan pada penelitian ini terdiri dari dua bagian yaitu data *training* dan data *testing*. Untuk melakukan pengujian terhadap hasil *training* maka perlu datanya data *testing*. Perbandingan data *training* dan *testing* yang diberikan adalah 80 : 20. Dengan demikian akan diambil 10 citra untuk masing-masing kategori yang nantinya akan diuji. Tabel 4 menunjukkan rincian perhitungan data *testing* setelah dilakukan pengujian.

Tabel 4. Rincian Perhitungan Akurasi Model Data Testing

No	Label	Benar	Salah	Total
1	Rossi	10	0	10
2	Dovizioso	10	0	10
3	Lorenzo	10	0	10
4	Pendrosa	8	2	10
5	Marquez	10	0	10
6	Vinales	10	0	10
Total		58	2	60
Akurasi		96,7%	3,33%	

D. Hasil Identifikasi Pembalap MotoGP

Setelah melalui tahap *training* menggunakan CNN, selanjutnya dilakukan *testing* terhadap model yang telah dihasilkan. Hasil dari proses *testing* adalah klasifikasi pembalap MotoGP beserta dengan derajat akurasi. Citra keluaran hasil *testing* ditunjukkan pada Gambar 6.



Gambar 6. Hasil Klasifikasi Pembalap MotoGP

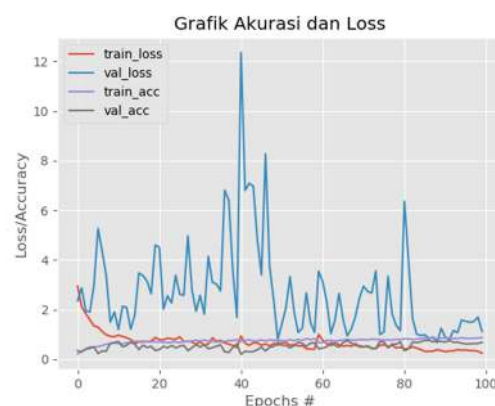
Berdasarkan hasil *training* dengan jumlah epoch sebanyak 150 dan *learning rate* 0,001 diperoleh hasil identifikasi pembalap MotoGP yang sangat akurat.

Gambar 6 menunjukkan bahwa program dapat mengidentifikasi pembalap MotoGP dengan benar dan dengan derajat kepercayaan yang tinggi yaitu diatas 90%.

IV. PEMBAHASAN

A. Analisis Hasil Training

Berdasarkan proses *training* yang dilakukan, diketahui bahwa mesin mampu melakukan pembelajaran terhadap data yang diolah. Gambar 7 menunjukkan grafik pergerakan nilai akurasi dan nilai *loss* untuk data *train* dan data *test* yang dihasilkan oleh setiap iterasi (*epoch*). Berdasarkan Gambar 7, diketahui pula bahwa nilai *loss* pada saat proses *training* mengalami penurunan. Hal ini baik, karena menandakan bahwa akurasi akan semakin besar.

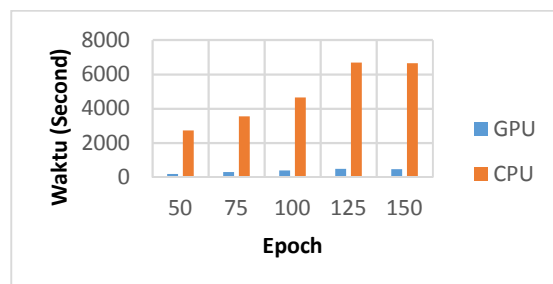


Gambar 7. Grafik Hasil Iterasi

Berbeda halnya dengan proses *testing*, proses penurunan nilai *loss* tidak signifikan terlihat. Sempat terjadi *loss* yang cukup besar pada epoch ke 40, namun semakin bertambahnya jumlah epoch, terjadi penurunan *loss* dan nilai akurasi semakin meningkat.

B. Perbandingan Training CPU dan GPU

Proses *training* merupakan tahap yang paling berat dalam klasifikasi citra. Oleh sebab itu diperlukan spesifikasi perangkat keras (prosesor atau gpu) yang besar. Tetapi lebih disarankan untuk menggunakan GPU dalam proses *training*, sebab waktu prosesnya akan jauh lebih cepat. Hal ini dibuktikan pada grafik perbandingan *training* menggunakan CPU dan GPU pada Gambar 8.



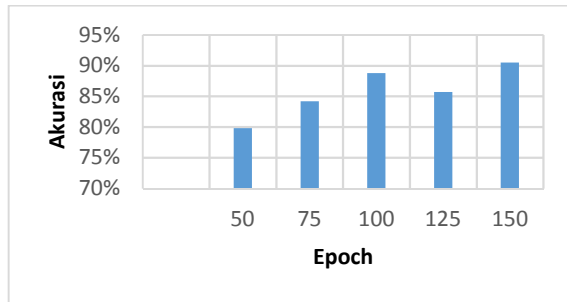
Gambar 8. Grafik Perbandingan Training CPU dan GPU

Gambar 8 menunjukkan bahwa penggunaan GPU dapat mempercepat waktu proses lebih cepat ±11 kali

lipat dibanding jika menggunakan CPU. Hal ini dikarenakan jumlah pemroses (*core*) pada GPU lebih banyak jika dibanding CPU.

C. Perbandingan Jumlah Iterasi (Epoch)

Jumlah epoch dapat mempengaruhi nilai akurasi. Hal ini terlihat pada Gambar 9 yang menunjukkan perbandingan nilai akurasi pada masing-masing *epoch*. Berdasarkan Gambar 9 diketahui bahwa semakin banyak jumlah epoch yang diberikan maka akan meningkatkan akurasi.



Gambar 9. Grafik Perbandingan Jumlah *Epoch*

Namun dalam beberapa kasus, terkadang semakin besar jumlah *epoch* tidak membuat akurasi naik. Hal ini terjadi pada jumlah *epoch* 125, dimana nilai akurasinya malah turun. Terdapat beberapa parameter yang mempengaruhi kondisi tersebut, diantaranya jumlah filter sebanyak 10 pada lapisan konvolusi pertama dan 20 pada lapisan kedua. Kemudian ukuran kernel pada lapisan pertama dan kedua sebesar 3x3.

V. PENUTUP

Berdasarkan hasil analisis yang telah dilakukan, diperoleh beberapa kesimpulan sebagai berikut.

1. Implementasi metode *Convolutional Neural Network* (CNN) mampu untuk melakukan klasifikasi objek pada citra. Hal ini ditunjukkan dengan akurasi data testing yang mencapai 96,7%.
2. Penggunaan GPU dalam proses training dapat meningkatkan kecepatan proses pembelajaran hingga ± 11 kali lipat dibandingkan CPU.
3. Terdapat beberapa faktor yang mempengaruhi nilai akurasi pada metode CNN, diantaranya jumlah iterasi (*epoch*) dan nilai *learning rate* yang diberikan.

DAFTAR PUSTAKA

- [1] I. W. S. E. Putra, "Klasifikasi Citra Menggunakan Convolutional Neural Network (Cnn) Pada Caltech 101," *J. Tek. ITS*, vol. 5, no. 1, pp. 65–69, 2016.
- [2] K. P. Danukusumo, "Implementasi Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Candi Berbasis GPU," 2017.
- [3] Salasabila, "Penerapan Deep Learning Menggunakan Convolutional Neural Network untuk Klasifikasi Citra Wayang Punakawan," 2018.
- [4] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," pp. 1–14, 2014.