

Klasifikasi Citra Multi-Kelas Menggunakan *Convolutional Neural Network*

Kamal Hasan Mahmud¹, Adiwijaya², Said Al Faraby³

^{1,2,3}Fakultas Informatika, Universitas Telkom, Bandung

¹kamalhm@student.telkomuniversity.ac.id, ²adiwijaya@telkomuniversity.ac.id,

³saidalfaraby@telkomuniversity.ac.id

Abstrak

ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) adalah suatu kompetisi tahunan yang melombakan klasifikasi basis data gambar yang ada pada ImageNet yang memiliki 1000 kelas dan setiap tahunnya terdapat berbagai macam arsitektur jaringan *convolutional neural network* (CNN) yang menjadi *state of the art*. CNN digunakan karena metode ini bekerja dengan sangat baik dengan volume data yang besar. Dari 1000 kelas, akan diambil 100 kelas yang akan dijadikan *dataset* penelitian untuk mengetahui strategi pembelajaran seperti apa pada CNN yang dapat memiliki performa terbaik dengan berbagai skenario. Skenario terbaik didapatkan dengan melakukan *training* dengan ukuran gambar yang kecil, lalu melakukan *training* kembali dengan memperbesar ukuran gambar. Skenario ini mendapatkan akurasi sebesar 75.82%, akurasi yang cukup tinggi untuk model yang dapat mengklasifikasikan 100 kelas. Skenario ini juga memiliki performa yang paling baik dalam klasifikasi keseluruhan berdasarkan ukuran evaluasi *confusion matrix*.

Kata kunci : *convolutional neural network*, multi-kelas, klasifikasi, gambar

Abstract

ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) is a yearly competition that competes on classifying ImageNet image database that has 1000 classes and every year, a new architecture of convolutional neural network (CNN) that becomes a state of the art emerge. CNN is used because this method works very well on a large volume of data. From 1000 classes available on ImageNet, 100 class will be used in this research dataset to know which learning strategy for CNN performed best in the varying scenario. The best scenario is achieved by training the image in smaller resolution then training it again with a larger resolution. This scenario achieved an accuracy of 75.82%, it's quite high for a model that can classify 100 class. It's also performed best according to the evaluation of the confusion matrix generated by each model.

Keywords: *convolutional neural network*, multi-class, classification, image

1. Pendahuluan

ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*) adalah kompetisi pengklasifikasian gambar yang diadakan oleh organisasi *ImageNet*, setiap diadakan kontes, selalu muncul berbagai macam jenis arsitektur CNN yang berbeda yang menjadi *state-of-the-art* untuk masalah klasifikasi [1]. ILSVRC 2012 menjadi pilihan penelitian karena *dataset* pada tahun ini sering digunakan dalam penelitian lain untuk menguji hasil klasifikasi dari suatu sistem yang telah dibuat [1].

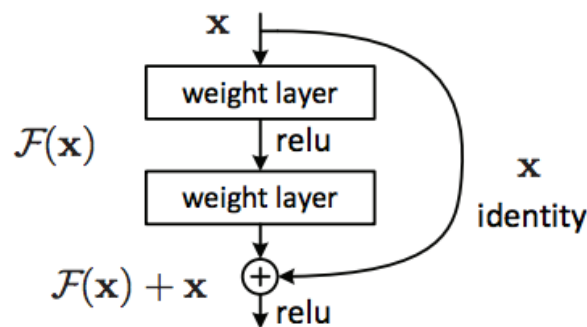
Convolutional neural network (untuk selanjutnya disebut CNN) adalah jenis jaringan saraf tiruan yang dapat terdiri dari beberapa lapisan, seperti *pooling layer*, *dense layer*, dan *fully connected layer*, oleh karena itu CNN memiliki variasi yang sangat banyak [3]. Selain CNN, ada metode-metode yang biasa digunakan untuk mengklasifikasikan suatu gambar antara lain adalah SVM (*Support Vector Machine*), dan KNN (*K-Nearest Neighbour*), setiap metode memiliki kekurangan dan kelebihan masing-masing. SVM dapat bekerja baik pada data linear, namun kesulitan dalam mengolah data berdimensi tinggi [2, 4]. KNN adalah metode yang sederhana, namun akan membutuhkan penyimpanan yang besar untuk menampung berjuta-juta objek yang ada pada *dataset* [2]. Oleh karena itu dipilhkan CNN yang dapat melakukan ekstraksi fitur, dapat bekerja dengan data yang berdimensi tinggi, dan memiliki performa yang bagus untuk masalah klasifikasi gambar [6, 9, 11]. Salah satu arsitektur yang cukup populer adalah *Residual Network* atau biasa disebut Resnet, arsitektur CNN ini dapat mengklasifikasikan gambar pada *dataset ImageNet* ILSVRC 2012 dengan tingkat akurasi yang tinggi, yaitu sebesar 80.62% untuk *top-1* yang didapatkan oleh salah satu arsitektur terbesar dari Resnet, yaitu ResNet152 [5].

2. Studi Terkait

2.1 *Residual Neural Network*

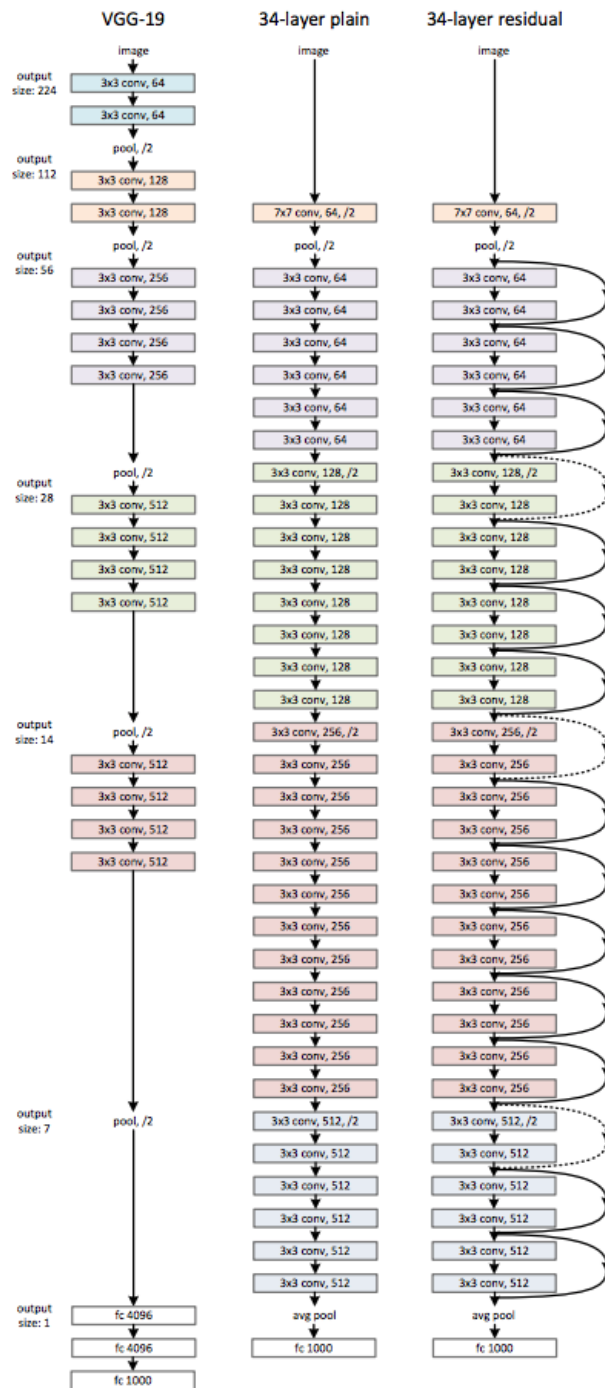
Residual neural network atau yang biasa disebut sebagai Resnet adalah salah satu jenis arsitektur yang cukup populer, arsitektur ini dibuat oleh Kaiming He et al [5]. Arsitektur ini cukup revolusioner pada saat itu karena arsitektur ini menjadi *state of the art* pada saat itu tidak hanya dalam klasifikasi, namun dalam semua kategori termasuk *object detection*, dan *semantic segmentation*. Arsitektur CNN yang memiliki kedalaman tinggi adalah salah satu hal penting dalam membangun model CNN yang memiliki performa yang baik, namun model CNN yang memiliki kedalaman yang tinggi juga memiliki masalah, yaitu *vanishing gradient problem*, yaitu suatu keadaan dimana hasil gradien yang dipelajari oleh model, tidak dapat mencapai *layer* pertama karena mengalami perkalian berkali-kali sehingga *layer* pertama tidak menerima gradien apa-apa, atau secara singkatnya, hal ini menyebabkan suatu CNN tidak dapat belajar dari *error* yang telah dikalkulasi [7]. Resnet memiliki berbagai macam jenis arsitektur, mulai dari 18, 34, 50, 101, sampai 152 *layer* [5]. Pada penelitian ini digunakan arsitektur Resnet34, yaitu arsitektur resnet yang memiliki 34 *layer*, arsitektur ini dipilih karena arsitektur ini memiliki performa yang baik pada kompetisi ILSVRC [5] dan karena keterbatasan *hardware* peneliti, maka arsitektur yang dipilih hanya memiliki 34 *layer*. Dari *dataset* keseluruhan, akan digunakan data *training* untuk melatih model CNN yang dibuat, dan data validasi untuk mengukur performa model yang dibuat. Performa dari sistem yang dibangun akan diukur dengan menggunakan metrik evaluasi akurasi, *training loss*, *validation loss* dan melihat matriks kekeliruan. Metrik ini digunakan juga pada berbagai penelitian yang dilakukan dengan CNN [10, 15, 13].

Hal yang diusung oleh Kaiming He et al. pada saat itu adalah dengan menggunakan sesuatu yang bernama *residual block*, blok ini adalah blok yang ada pada tiap lapis arsitektur CNN Resnet dan menjadi fundamental dari arsitektur tersebut, gambaran dari blok ini dapat dilihat pada Gambar 1.



Gambar 1. Blok residual yang menjadi fondasi dari arsitektur Resnet [5].

Blok ini menambahkan suatu jalan pintas yang berfungsi sebagai fungsi identitas, yang secara tidak langsung akan melewati proses *training* untuk satu *layer* atau lebih, sehingga membuat sesuatu yang bernama *residual block*.



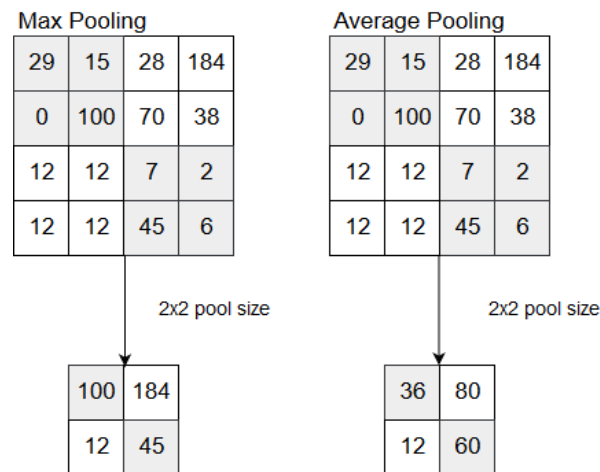
Gambar 2. Contoh arsitektur keseluruhan dari Resnet [5].

Penggunaan *residual block* dapat terlihat pada Gambar 2, dimana arsitektur VGGnet berada pada paling kiri, arsitektur yang tanpa menggunakan *residual block* terdapat pada bagian tengah, dan arsitektur yang menggunakan *residual block* terdapat pada bagian paling kanan. Dapat dilihat bahwa dengan melewati beberapa layer, arsitektur ini dapat mengatasi permasalahan *vanishing gradient problem* karena gradien dapat sampai kepada layer pertama tanpa harus melewati jumlah perkalian yang sama dibandingkan dengan arsitektur yang sama tanpa menggunakan *residual block*.

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n) \tag{1}$$

Pada baris pertama, hasil konvolusi akan membentuk *filter* sebanyak 64 buah, operasi konvolusi dapat dilihat pada rumus (1) dimana setiap elemen yang pada matriks *I* dan *K* mengalami perkalian matriks sehingga menghasilkan filter *S*. Pada tahap ini model belum memasuki modul *residual*. Setelah tahap ini selesai, gambar akan

memasuki *pooling layer*, *pooling layer* adalah *layer* yang bertugas untuk mengurangi resolusi dari suatu gambar yang telah diproses, *pooling layer* berfungsi untuk mengurangi *noise* yang ada dalam gambar tersebut, terdapat dua jenis *pooling* yang ada, yaitu *max pooling* dan *average pooling*, ilustrasi dari operasi ini dapat dilihat pada Gambar 3, *average pooling* lebih umum digunakan pada berbagai macam arsitektur CNN [10, 15, 8, 13].



Gambar 3. Ilustrasi operasi *max pooling* dan *average pooling* pada CNN[3].

Lalu tiga modul *residual* ditumpuk diatas masing-masing *layer*, dan setiap *layer* akan mempelajari 128 filter untuk setiap proses *konyolusi*. Dimensi spasial dari gambar akan berkurang, lalu akan ditumpuk kembali 4 buah modul *residual*, yang dimana setiap modul akan mempelajari 256 filter. Tahap terakhir, kita akan mengurangi dimensi spasialnya kembali dan masuk kepada tahap menumpuk 6 buah modul *residual*, dimana setiap layer akan mempelajari 512 filter. Hasil dari setiap filter ini akan melewati *average pooling* dan masuk ke dalam jaringan *fully connected* dengan fungsi aktivasi *softmax* untuk diketahui hasil klasifikasi.

$$\text{softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^n \exp(x_j)} \quad (2)$$

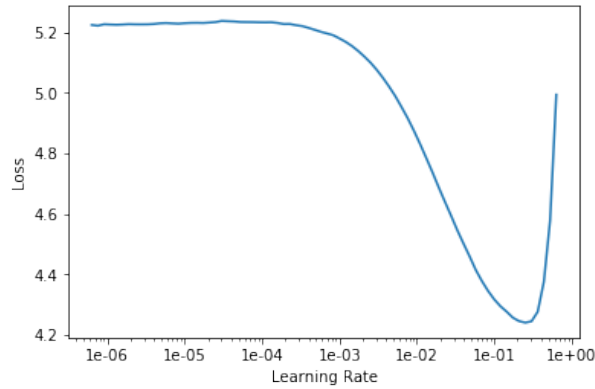
Softmax adalah suatu fungsi yang biasanya digunakan untuk menghitung probabilitas yang umum digunakan untuk melakukan klasifikasi multi kelas, nilai softmax berada pada *range* 0 - 1 dan memiliki jumlah 1 jika seluruh elemennya dijumlahkan [3]. Fungsi ini biasanya digunakan di ujung *layer* dari *fully connected layer* yang digunakan pada CNN untuk menghasilkan nilai probabilitas suatu objek terhadap kelas yang ada. Rumus softmax dapat dilihat pada (2), dimana kita akan menghitung nilai eksponen dari x_i terhadap jumlah dari seluruh nilai eksponen yang ada.

Untuk suatu model dapat memperbaiki dari kesalahan yang dibuat pada saat *training*, digunakan fungsi *cross entropy loss* (3), dimana fungsi ini akan menghitung kelas aktual t_i berbanding dengan s_i yang akan menghitung jumlah log negatif dari hasil perkalian nilai prediksi dikali kelas asli lalu dibagi dengan jumlah kelas.

$$CE = -\sum t_i \log(s_i) \quad (3)$$

2.2 Learning Rate

Learning rate adalah salah satu *hyper parameter* yang sangat mempengaruhi performa suatu model CNN. Dalam melakukan pencarian *learning rate* yang tepat, digunakan suatu metode yang bernama *cyclical learning rates* [14], dalam metode ini, akan dilakukan beberapa *epochs training*, dengan *learning rate* yang dimulai dari nilai yang kecil, dan setiap iterasi akan diperbesar *learning rate*-nya, setiap iterasi akan dilihat *loss* yang didapat, dan jika *loss* yang didapat meningkat cukup drastis maka proses pencarian akan diberhentikan. Berdasarkan data *loss* yang didapat, maka dipilih *learning rate* sesaat sebelum *loss* memiliki nilai terendah.



Gambar 4. Plot *learning rate* terhadap *loss*.

Dapat dilihat pada Gambar 3 bahwa pada saat *learning rate* berada pada titik $1e - 02$ titik ini memiliki nilai *loss* terkecil kedua, maka titik ini cocok untuk dipilih untuk menjadi *learning rate* [14].

2.3 Augmentasi Data

Augmentasi data adalah suatu proses dalam pengolahan data gambar, augmentasi merupakan proses mengubah atau memodifikasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda, namun manusia masih dapat mengetahui bahwa gambar yang diubah tersebut adalah gambar yang sama [12]. Augmentasi dapat meningkatkan akurasi dari model CNN yang dilatih karena dengan *augmentasi* model mendapatkan data-data tambahan yang dapat berguna untuk membuat model yang dapat melakukan generalisasi dengan lebih baik. Augmentasi yang dilakukan pada penelitian ini adalah membalikan gambar secara horizontal, melakukan *zoom-in* secara acak, dengan maksimal *zoom* sebesar 50% dari besar gambar, dan juga melakukan rotasi gambar secara acak dengan derajat maksimal 90° . Salah satu jenis augmentasi yang umum dilakukan adalah dengan melakukan perputaran gambar dengan besar tertentu, contoh dari penggunaan augmentasi ini dapat dilihat pada Gambar 4.



Gambar 5. Augmentasi dengan perputaran gambar maksimal sebesar 90° .

2.4 ImageNet

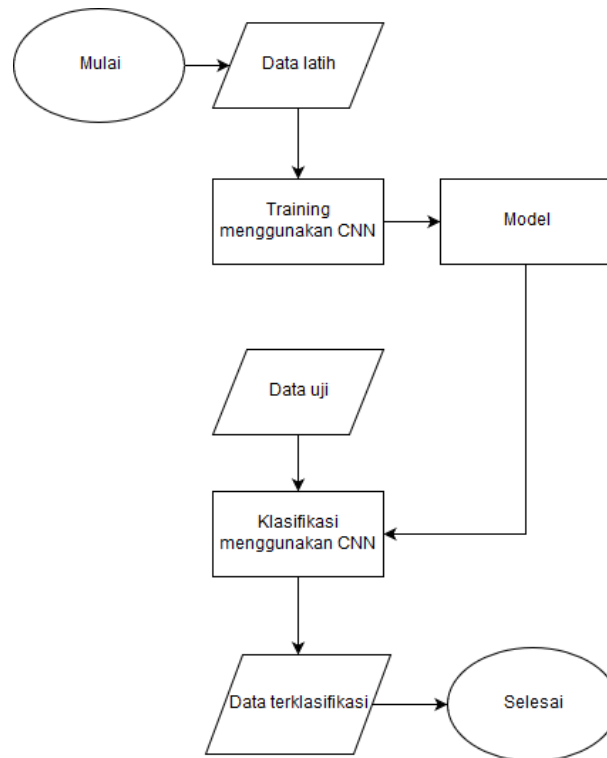
Dataset yang digunakan pada penelitian ini adalah *dataset* ImageNet ILSVRC 2012 yang terdiri dari 1000 kelas, namun karena keterbatasan *hardware* maka pada penelitian ini hanya akan diambil 100 kelas pertama dari 1000 kelas yang tersedia. Contoh dari *dataset* ini dapat dilihat pada Gambar 5, *dataset* ini berisi berbagai jenis binatang.



Gambar 6. Contoh beberapa gambar pada dataset yang digunakan.

3. Sistem yang Dibangun

Pada penelitian ini dibangun sebuah model *convolutional neural network* yang dapat mengklasifikasikan gambar-gambar yang berasal dari *dataset* Imagenet ILSVRC 2012. Diagram alir dari sistem yang dibangun dapat dilihat pada Gambar 7.



Gambar 7. Diagram alir sistem klasifikasi gambar.

Pada tahap *training* akan dilakukan pelatihan model CNN dengan arsitektur Resnet34, sebelum memulai *training*, akan dilakukan algoritma pencarian *learning rate* yang sudah dijelaskan pada Bab 2.2, algoritma ini dijalankan setiap melakukan training telah selesai untuk mengevaluasi *learning rate* baru yang harus diubah menyesuaikan dengan *loss* yang didapatkan oleh model. Gambar yang ada pada data latih akan diproses oleh model CNN dimana setiap gambar yang ada akan melewati setiap bagian dari arsitektur Resnet34 yang sudah dijelaskan pada Bab 2.1, keluaran dari proses ini adalah hasil prediksi dari gambar-gambar yang masuk kedalam model tersebut. Setiap *epochs* yang dilakukan, model akan mengevaluasi juga dataset yang ada pada data validasi untuk mengetahui seberapa baik model dapat mengklasifikasikan gambar yang belum dilihat sebelumnya pada saat *training*. Hasil *training* model terbaik akan dievaluasi performanya dan akan dilakukan analisis. Untuk mengetahui performa dari model arsitektur CNN yang dibangun dalam mengklasifikasikan gambar, dan mengevaluasi hasil eksperimen maka akan dibuat 3 skenario utama. Pertama adalah skenario menggunakan gambar berukuran 64x64 *pixel*, resolusi

ini adalah resolusi dengan ukuran yang lebih kecil daripada resolusi gambar standar yang digunakan pada model CNN. Kedua adalah skenario dengan menggunakan gambar berukuran 224×224 pixel, resolusi ini adalah resolusi gambar yang digunakan oleh berbagai macam arsitektur dalam melakukan klasifikasi ImageNet [13, 5]. Ketiga adalah dengan menggunakan gambar berukuran lebih kecil, yaitu 64×64 , lalu dilatih kembali menggunakan ukuran 224×224 pixel. Dari 3 skenario utama tersebut, setiap skenario akan dilakukan *training* dengan augmentasi dan tanpa augmentasi.

4. Evaluasi

Dengan menggunakan augmentasi data, akurasi yang didapat ternyata lebih kecil dibandingkan dengan yang tanpa menggunakan augmentasi, perbandingan akurasi dapat dilihat pada Gambar 8, dimana dengan menggunakan augmentasi, diperoleh akurasi sebesar 49.32% untuk gambar berukuran 64×64 dibandingkan dengan 50.52%. *Error rate* yang didapatkan oleh model tanpa augmentasi juga lebih kecil dibandingkan dengan yang menggunakan augmentasi, seperti yang dapat dilihat pada Gambar 9. Meskipun augmentasi mendapatkan performa akurasi yang lebih kecil, namun performa model cukup baik secara keseluruhan, dapat dilihat pada Tabel 1 dan Tabel 2 bahwa walaupun augmentasi memiliki akurasi yang lebih kecil, jumlah kelas yang keliru lebih sedikit, dan cenderung lebih merata.

Tabel 1. Kekeliruan 5 kelas teratas hasil prediksi pada model dengan ukuran gambar 64×64

Nomor	Kelas		Jumlah Kekeliruan
1	European Green Lizard	Carolina Anole	18
2	Partridge	Ruffed Grouse	17
3	Smooth Green Snake	Green Mamba	16
4	Saharan Horned Viper	Sidewinder	15
5	European Garden Spider	Barn Spider	14

Tabel 2. Kekeliruan 5 kelas teratas hasil prediksi pada model dengan ukuran gambar 64×64 dengan augmentasi.

Nomor	Kelas		Jumlah Kekeliruan
1	Tiger Shark	Hammerhead Shark	20
2	Carolina Anole	European Green Lizard	12
3	Saharan Horned Viper	Sidewinder	12
4	Bald Eagle	Kite	12
5	Great White Shark	Hammerhead Shark	12

Dengan menggunakan ukuran gambar yang diperbesar, yaitu dengan ukuran 224×224 pixel, didapatkan akurasi yang lebih tinggi dikarenakan informasi yang dapat dipelajari lebih banyak, dengan menggunakan ukuran gambar 224×224 , diperoleh akurasi sebesar 77.12%, dan pada Tabel 3 dapat dilihat kekeliruan kelas hasil prediksi oleh model tersebut. Dapat dilihat bahwa jumlah kekeliruan teratas yang lebih tinggi daripada gambar yang berukuran kecil, menandakan bahwa model ini mengalami *overfitting*. Sama halnya dengan kasus 64×64 , model berukuran 224×224 jika menggunakan augmentasi maka akurasi yang didapat tidak mendapati akurasi setinggi model yang tidak menggunakan augmentasi, yaitu hanya sebesar 75.8%. Perbandingan dari kedua model ini dapat dilihat pada Gambar 11. Perbandingan kekeliruan kelas yang diprediksi oleh model berukuran 224 dapat dilihat pada Tabel 3 dan Tabel 4, ternyata augmentasi juga tidak selalu memberikan hasil yang baik, tidak hanya akurasinya lebih kecil, namun kelas yang keliru juga didapatkan lebih banyak pada model yang menggunakan augmentasi. Hal ini kemungkinan terjadi karena augmentasi yang dilakukan cukup kompleks namun model yang dilatih kurang kompleks

Hal ini juga terjadi pada gambar yang berukuran 224×224 , skenario yang menggunakan augmentasi memiliki akurasi yang lebih kecil dibandingkan dengan tanpa augmentasi, dimana tanpa augmentasi mendapatkan akurasi sebesar 77.12%, dan 75.8% tanpa augmentasi.

Tabel 3. Kekeliruan 5 kelas teratas hasil prediksi pada model dengan ukuran gambar 224x224

Nomor	Kelas Target	Kelas Prediksi	Jumlah Kekeliruan
1	Carolina Anole	European Green Lizard	29
2	Sidewinder	Saharan Horned Viper	27
3	Terrapin	Mud Turtle	23
4	Smooth Green Snake	Green Mamba	16
5	Stingray	Electric Ray	16

Tabel 4. Kekeliruan 5 kelas teratas hasil prediksi pada model dengan ukuran gambar 224x224 dengan augmentasi

Nomor	Kelas Target	Kelas Prediksi	Jumlah Kekeliruan
1	Yellow Garden Spider	European Garden Spider	42
2	European Green Lizard	Carolina anole	24
3	Sidewinder	Saharan horned viper	24
4	African Rock Python	Boa Constrictor	22
5	Terrapin	Mud Turtle	18

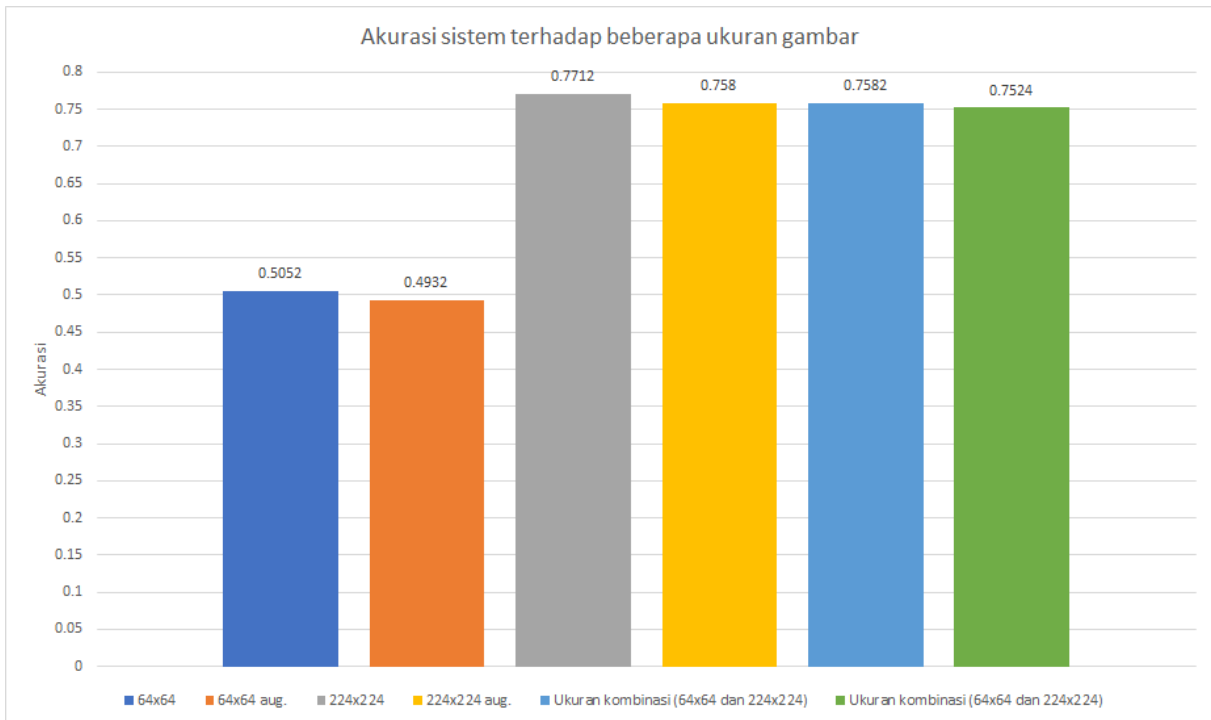
Selain kedua skenario tersebut, dilakukan suatu skenario dimana model akan dilatih menggunakan data yang berukuran 64x64 pixel, lalu akan dilatih kembali menggunakan gambar yang sama, namun dengan gambar berukuran 224x224 pixel, dengan skenario ini didapatkan akurasi sebesar 75.82% dan jumlah kelas yang keliru dapat dilihat pada Tabel 5. Hasil skenario yang sama namun dengan augmentasi dapat dilihat pada Tabel 6.

Tabel 5. Kekeliruan 5 kelas teratas hasil prediksi pada model dengan ukuran gambar 64x64, lalu diperbesar menjadi 224x224

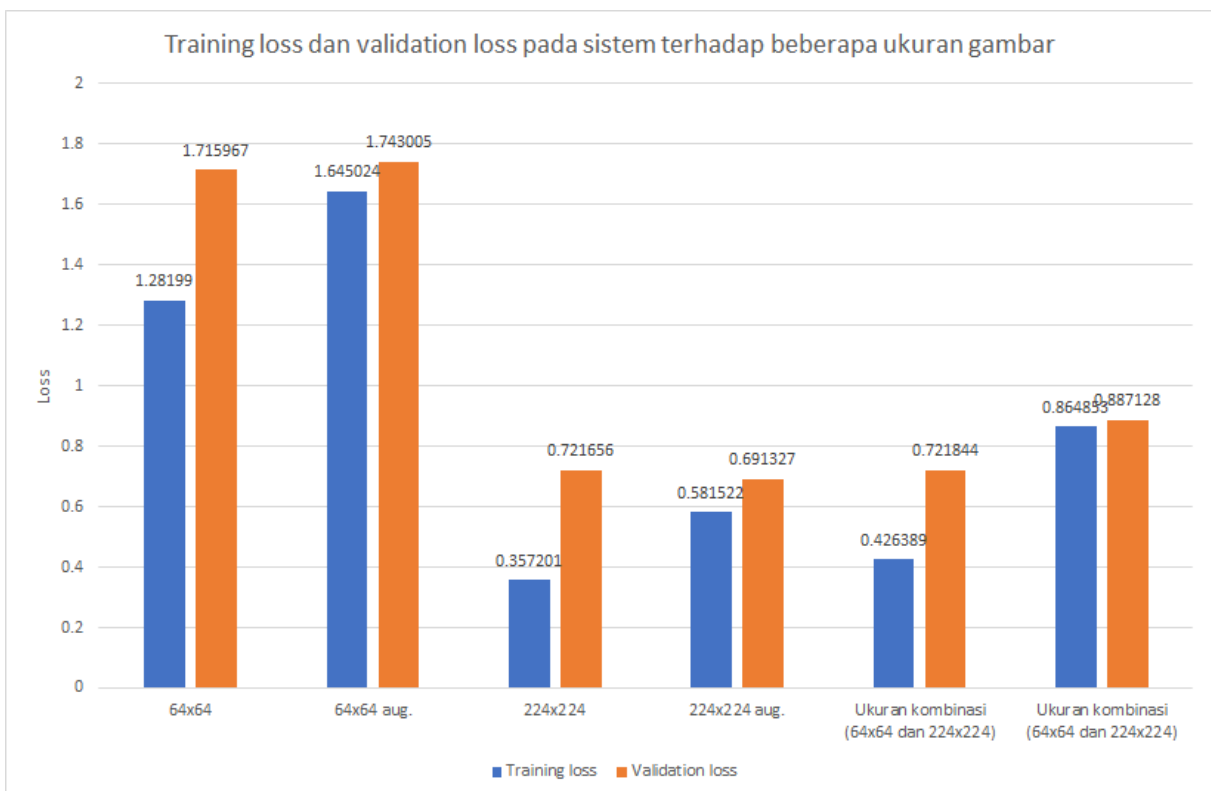
Nomor	Kelas Target	Kelas Prediksi	Jumlah Kekeliruan
1	Great White Shark	Tiger Shark	11
2	Terrapin	Mud Turtle	11
3	Tiger Shark	Hammerhead Shark	11
4	Goldfinch	Bee Eater	10
5	Terrapin	Box Turtle	10

Tabel 6. Kekeliruan 5 kelas teratas hasil prediksi pada model dengan ukuran gambar 64x64, lalu diperbesar menjadi 224x224 dengan augmentasi

Nomor	Kelas Target	Kelas Prediksi	Jumlah Kekeliruan
1	Cock	Hen	20
2	Sidewinder	Saharan Horned Viper	19
3	European Garden Spider	Yellow Garden Spider	17
4	Barn Spider	European Garden Spider	17
5	Bald Eagle	Kite	16



Gambar 8. Akurasi yang didapat dari 6 skenario yang dilakukan.



Gambar 9. Training dan validation loss yang didapat dari 6 skenario yang dilakukan.

5. Kesimpulan

Dalam penelitian ini dilakukan pengklasifikasian gambar multi-kelas pada dataset ImageNet yang telah dilaksanakan dalam penelitian ini, digunakan arsitektur Resnet34. Menggunakan metode *cyclical learning rate* untuk

mencari learning rate dimana didapatkan learning rate awal sebesar $1e - 2$ dan nilai ini akan dikurangi setelah akurasi dari model mengalami saturasi. Berdasarkan skenario yang dilakukan, didapatkan akurasi terbaik tanpa menggunakan augmentasi dengan akurasi sebesar 75.82%. Namun, hasil evaluasi dari matriks kekeliruan menunjukkan bahwa model yang tidak menggunakan augmentasi dan melakukan perubahan ukuran gambar memiliki kekeliruan matriks terbaik dengan 11 kekeliruan tertinggi, walaupun akurasi-nya sedikit lebih kecil yaitu 75.24%. Dapat ditarik beberapa kesimpulan bahwa 1) augmentasi data, dengan menggunakan augmentasi data, model akan menerima lebih banyak data, tetapi jika tidak diimbangi dengan *tuning* yang baik, maka model dapat memiliki performa yang buruk. 2) Memperkecil lalu memperbesar ukuran gambar yang dilatih dapat membuat model dapat memprediksi dengan lebih merata meskipun akurasi yang didapat sama. Untuk penelitian selanjutnya, diharapkan untuk menggunakan arsitektur yang lebih besar, dan jumlah kelas yang lebih banyak lagi untuk memastikan bahwa skenario ini dapat dilakukan pada ukuran sistem yang lebih besar.

Daftar Pustaka

- [1] Imagenet Large Scale Visual Recognition Challenge (ILSVRC). Diakses 9 April 2018.
- [2] S. R. Amendolia, G. Cossu, M. Ganadu, B. Golosio, G. Masala, and G. M. Mura. A comparative study of k-nearest neighbour, support vector machine and multi-layer perceptron for thalassemia screening. *Chemo-metrics and Intelligent Laboratory Systems*, 69(1-2):13–20, 2003.
- [3] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [4] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.
- [5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv:1512.03385 [cs]*, Dec 2015. arXiv: 1512.03385.
- [6] B. Hu, Z. Lu, H. Li, and Q. Chen. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050, 2014.
- [7] Y. Hu, A. E. G. Huber, J. Anumula, and S. Liu. Overcoming the vanishing gradient problem in plain recurrent networks. *CoRR*, abs/1801.06105, 2018.
- [8] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167 [cs]*, Feb 2015. arXiv: 1502.03167.
- [9] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):221–231, 2013.
- [10] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, May 2017.
- [11] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov. 1998.
- [12] L. Perez and J. Wang. The effectiveness of data augmentation in image classification using deep learning, 2017.
- [13] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556 [cs]*, Sep 2014. arXiv: 1409.1556.
- [14] L. N. Smith. Cyclical learning rates for training neural networks. *CoRR*, abs/1506.01186, 2015.
- [15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv:1409.4842 [cs]*, Sep 2014. arXiv: 1409.4842.

Lampiran