

Perbandingan *Convolutional Neural Network* pada *Transfer Learning Method* untuk Mengklasifikasikan Sel Darah Putih

Daniel Martomanggolo Wonohadidjojo

Program Studi Informatika, Fakultas Teknologi Informasi, Universitas Ciputra Surabaya, Surabaya, Indonesia
daniel.m.w@ciputra.ac.id

Diterima 17 Mei 2021

Disetujui 13 Juni 2021

Abstract—Analysis of White Blood Cells structure from microscopic images and classification of cells into types is challenging. Although white blood cells can be differentiated based on their shape, color and size, one challenging aspect is that they are surrounded by other blood components such as red blood cells and platelets. In this study, *transfer learning* method using four *network* architectures that have been trained in advance is applied to classify the white blood cell images. The *network* architectures used are AlexNet, GoogleNet, ResNet-50 and Visual Geometry Group (VGG) - 16. A comparative analysis of the performance of these architectures was carried out in classifying the images. The evaluation method was undertaken using *Confusion matrix*. The *performance metrics* measured in the evaluation are *Accuracy*, *Precision*, *Recall* and *Fmeasure*. The results showed that all architectures succeeded in classifying white blood cells using the *transfer learning* method. ResNet-50 is the *network* architecture that shows the highest performance in classifying white blood cell images.

Index Terms—white blood cells, *transfer learning*, AlexNet, GoogleNet, ResNet-50, VGG-16

I. PENDAHULUAN

Darah terutama mengandung tiga jenis sel: sel darah merah, sel darah putih dan trombosit. Dari tiga jenis tersebut, sel darah putih adalah jenis sel darah yang dibuat di sumsum tulang dan ditemukan di darah dan jaringan getah bening. Sel darah putih merupakan bagian dari sistem kekebalan tubuh. Mereka membantu tubuh melawan infeksi dan penyakit lainnya. Sel darah putih dapat dikategorikan menjadi dua jenis, ditentukan oleh penampilan sitoplasma. Jenis pertama adalah Granulosit dan termasuk Basophil, *Eosinophil* dan *Neutrophil*. Kelompok kedua, yang disebut Agranulosit, termasuk *Lymphocyte* dan *Monocyte* [1].

Sel darah putih memiliki fungsi penting untuk sistem kekebalan tubuh, karena sel darah putih merupakan pertahanan utama tubuh terhadap infeksi dan penyakit. Memeriksa jumlah sel darah putih dalam darah biasanya merupakan bagian dari tes sel darah lengkap. Ini dapat digunakan untuk mencari kondisi

seperti infeksi, peradangan, alergi, dan leukemia. Hal ini merupakan masalah tersendiri karena kategori sel darah putih mencakup sel-sel yang dapat sangat bervariasi satu sama lain. Meskipun sel darah putih dapat dibedakan berdasarkan bentuk dan ukurannya, satu aspek yang menantang adalah sel darah putih dikelilingi oleh komponen darah lain seperti sel darah merah dan trombosit.

Berdasarkan penjelasan tersebut, maka diperlukan metode untuk mengklasifikasikan sel darah putih secara akurat.

Deep Learning menggunakan *Convolution Neural Networks* (CNN) [2] saat ini merupakan pilihan terbaik dalam aplikasi pencitraan medis seperti deteksi dan klasifikasi [3]. Meskipun CNN mencapai hasil terbaik pada kumpulan data yang besar, tapi CNN membutuhkan banyak data dan sumber daya komputasi untuk dilatih. Dalam banyak kasus, kumpulan data terbatas dan mungkin tidak cukup untuk melatih CNN dari awal. Dalam skenario seperti itu, untuk memanfaatkan kekuatan CNN dan pada saat yang sama mengurangi sumber daya komputasi, *transfer learning* dapat digunakan [4]. Dalam pendekatan ini, CNN awalnya dilatih pada kumpulan data citra umum yang besar dan beragam dan kemudian diterapkan ke tugas tertentu.

Terdapat beberapa *neural network* pra-terlatih yang telah memenangkan kompetisi internasional seperti VGGNet [5], ResNet [6], GoogleNet [7], dan AlexNet [8]. Dalam [9] dilakukan evaluasi arsitektur CNN yang berbeda, dan *transfer learning* mencapai kinerja skor tertinggi pada kelenjar getah bening torako-abdominal serta klasifikasi penyakit paru interstitial.

Beberapa studi telah dilakukan untuk menerapkan CNN pada klasifikasi citra sel darah putih. Ekstraksi fitur dari citra sel darah putih menggunakan VGGNet, yang kemudian difilter menggunakan *Salp Swarm Algorithm* (SESSA) yang ditingkatkan secara statistik dilakukan pada [1]. Pada [10] diterapkan klasifikasi CNN untuk mengeksplorasi kelayakan pendekatan *deep learning* untuk mengidentifikasi limfosit dan

subtipe acute *lymphoblastic leukaemia*, dan dibandingkan dengan pendekatan *Support Vector Machines* (SVM). Selain itu, dua klasifikasi pembelajaran mesin tradisional, *multilayer perceptron* (MLP), dan *random forest* juga diterapkan untuk perbandingan.

Suatu model *Deep Learning* berbasis *Hybrid CNN* untuk mengklasifikasikan leukosit eosinofil, limfosit, monosit, dan *neutrophil* diusulkan pada [11]. Model yang disajikan tersebut didasarkan pada arsitektur AlexNet dan GoogleNet yang telah dilatih sebelumnya. Fitur Vektor pada *pooling layer* terakhir pada kedua arsitektur CNN telah digabungkan, dan kemudian fitur vektor yang dihasilkan diklasifikasikan oleh SVM.

Pada [12][13] dilakukan kombinasi beberapa arsitektur *deep learning* untuk meningkatkan kegunaan *transfer learning* dalam mengklasifikasikan citra berbasis sel.

Berdasarkan seluruh penjelasan tersebut maka pada penelitian ini digunakan *transfer learning* untuk melatih ulang CNN untuk mengklasifikasikan citra sel darah putih.

Selanjutnya paper ini disusun sebagai berikut: section II menjelaskan tentang kajian Pustaka yang digunakan pada penelitian ini, section III menjelaskan tentang metode penelitian yang digunakan, section IV menjelaskan tentang hasil uji coba dan eksperimen serta pembahasannya dan section V menjelaskan tentang kesimpulan penelitian ini.

II. KAJIAN PUSTAKA

A. Transfer Learning

CNN yang digunakan untuk penerapan *transfer learning*, sebelumnya telah di-*training* dengan menggunakan dataset citra sejumlah obyek/ benda. Pada *training* tersebut, *network* dapat mengklasifikasikan citra ke dalam sejumlah kategori obyek, seperti *keyboard*, cangkir kopi, pensil, dan banyak hewan. Dengan demikian *network* tersebut telah mempelajari representasi fitur yang kaya untuk berbagai citra. Selanjutnya *network* mengambil citra sebagai input dan kemudian mengeluarkan label untuk tiap obyek dalam citra bersama dengan probabilitas untuk masing-masing kategori obyek.

Transfer learning merupakan metode yang menggunakan *network* yang sudah dilatih sebelumnya dan menggunakannya sebagai titik awal untuk mempelajari tugas baru. Melakukan fine tuning *network* dengan *transfer learning* jauh lebih cepat dan lebih mudah daripada melatih *network* dari awal dengan bobot yang diinisialisasi secara acak.

Transfer learning ini dapat mentransfer fitur yang dipelajari ke tugas baru dengan menggunakan lebih sedikit dataset pada proses *training* sehingga prosesnya lebih cepat. Melakukan *fine tuning network*

biasanya lebih lambat dan membutuhkan lebih banyak upaya daripada ekstraksi fitur sederhana. Tetapi jika digunakan *transfer learning* maka karena *network* sudah di-*training* sebelumnya dan mempelajarinya untuk mengekstrak serangkaian fitur yang berbeda, maka *network* yang baru menjadi lebih akurat. Dengan demikian maka fine-tuning pada *network* yang baru dapat bekerja lebih baik daripada ekstraksi fitur meskipun hanya menggunakan dataset dalam jumlah yang lebih kecil, karena *network* yang baru tersebut telah belajar untuk mempelajari fitur yang baru.

B. Evaluasi performa

Untuk mengukur performa model *network* tersebut dalam melakukan klasifikasi digunakan *Confusion matrix* (CM). Pada CM nilai prediksi dibandingkan dengan nilai aktual. Nilai prediksi adalah keluaran dari program dimana nilainya Positif dan Negatif sedangkan nilai aktual adalah nilai sebenarnya yang berupa nilai True dan False. Dengan demikian terdapat 4 kemungkinan hasil prediksi yaitu True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN). Berdasarkan kemungkinan tersebut maka *performance metrics* yang digunakan direpresentasikan pada persamaan (1) – (5) [14][15].

Accuracy merupakan *metric* yang banyak digunakan pada evaluasi performa tersebut. *Metric* ini menghitung persentase sampel yang diklasifikasikan dengan benar, dan direpresentasikan pada Persamaan (1).

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \times 100 \quad (1)$$

Precision mengukur seberapa "tepat" model tersebut berdasarkan pada yang diprediksi positif dan berapa banyak di antaranya yang sebenarnya positif. *Metric* ini dihitung menggunakan persamaan (2).

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Sensitivity atau *Recall* merupakan *metric* yang mengukur kesesuaian dengan keakuratan contoh positif, dan mengacu pada berapa banyaknya contoh kelas positif diberi label dengan benar. *Metric* ini dihitung dengan persamaan (3), di mana TP mengacu pada true positive, yang merupakan jumlah kejadian yang positif dan diidentifikasi dengan benar, dan FN mewakili *false negative*, yang merupakan jumlah kasus positif yang salah diklasifikasikan sebagai negatif.

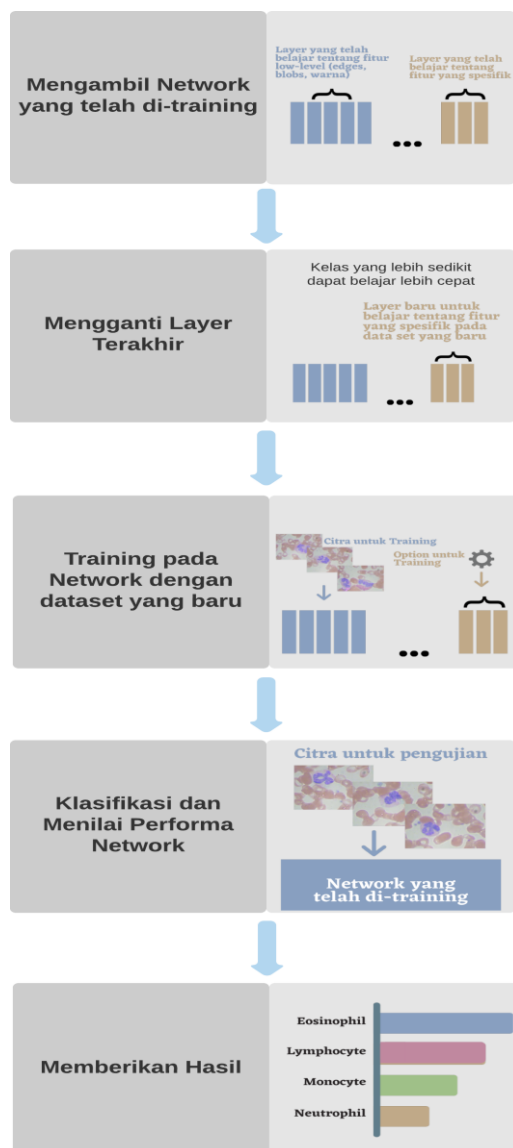
$$Recall = \frac{TP}{TP+FN} \quad (3)$$

Fmeasure merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan. *Metric* ini dihitung menggunakan persamaan (4).

$$F \text{ measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

III. METODE PENELITIAN

Network terlatih yang digunakan pada *transfer learning* ini adalah model CNN. CNN sama dengan *neural network* lainnya, terdiri dari fungsi bobot, bias, dan aktivasi. CNN memiliki 2 bagian besar *layer* yaitu *layer* untuk *feature extraction* dan *layer* untuk klasifikasi. *Layer* untuk *feature extraction* terdiri dari *layer convolutional*, *pooling layer*, *stride*, dan *padding*. Sedangkan *layer* untuk klasifikasi terdiri dari *fully connected layer*, *softmax*, dan *layer* keluaran. *Network* terlatih inilah yang digunakan pada metode *transfer learning* pada penelitian ini. Tahap *transfer learning* yang digunakan pada penelitian ini meliputi: meng-*import network* yang sudah dilatih sebelumnya, mengganti *layer* klasifikasi, melatih *network* pada data citra sel darah putih, mengklasifikasikan sel darah putih dan mendapatkan ukuran kinerja yang akurat. Gambar 1 menunjukkan metode *transfer learning* yang digunakan pada penelitian ini.



Gambar 1. Metode *Transfer learning* yang digunakan

A. Meng-import Network Yang Telah Dilatih

Pada penelitian ini arsitektur *network* yang digunakan adalah AlexNet, GoogleNet, ResNet-50, dan VGG-16 sebagai *network* yang telah dilatih. Setiap *network* terlatih memiliki konfigurasi *layer* yang berbeda. *Layer* di awal dan tengah disebut *layer feature extraction*. *Layer* tersebut menghasilkan fitur sederhana seperti *brightness* dan *edge*, hingga fitur kompleks yang unik seperti warna dan bentuk. Hasil pembelajaran pada *layer feature extraction* pada lingkungan sumber (lingkungan dimana *network* tersebut berasal) dapat ditransfer ke *layer feature extraction* pada lingkungan target (lingkungan baru pada penelitian ini). Pembelajaran *feature extraction* pada lingkungan target digunakan untuk meng-ekstrak citra data sel darah putih yang digunakan pada tahap *training*.

Arsitektur GoogLeNet memiliki 22 *layer* jika dihitung hanya *layer* yang memiliki parameter (atau 27 lapisan jika kita juga menghitung *pooling*). *Network* ini pada lingkungan asalnya telah dilatih pada lebih dari satu juta citra dan dapat mengklasifikasikan citra ke dalam 1000 kategori obyek. Arsitektur *network* ini telah mempelajari representasi fitur yang kaya untuk berbagai citra. *Network* mengambil citra sebagai input dan output label untuk obyek dalam citra bersama-sama dengan probabilitas untuk setiap kategori obyek [7].

Arsitektur AlexNet berisi delapan *layer* dimana lima yang pertama adalah *layer convolutional*, beberapa di antaranya diikuti oleh *layer max-pooling*, dan tiga yang terakhir adalah *layer fully-connected*. Arsitektur *network* ini menggunakan fungsi aktivasi ReLU non-jenuh, yang menunjukkan peningkatan kinerja *training* di atas tanh dan sigmoid [8].

Arsitektur ResNet-50 adalah Jaringan *Residual* yang terdiri dari 50 *layer*. Seperti yang ditunjukkan oleh namanya, *network* ini menggunakan pembelajaran *Residual*. Pada pembelajaran *Residual*, *network* bukannya mencoba mempelajari beberapa fitur, tetapi *network* ini mempelajari beberapa *Residual*. *Residual* secara sederhana dapat dipahami sebagai pengurangan fitur yang dipelajari dari masukan suatu *layer*. ResNet melakukan ini dengan menggunakan koneksi pintasan yaitu langsung menghubungkan input dari *layer* ke- n ke beberapa *layer* berikutnya ($n + x$). Telah dibuktikan bahwa melatih bentuk *network* ini lebih mudah daripada melatih *deep convolutional neural networks* yang sederhana dan juga masalah penurunan akurasi dapat teratasi [6].

VGG16 adalah arsitektur CNN yang digunakan untuk memenangkan kompetisi ILSVR (Imagenet) pada tahun 2014. Ini dianggap sebagai salah satu arsitektur model vision yang sangat baik hingga saat ini. Hal yang paling unik tentang VGG16 adalah bahwa alih-alih memiliki sejumlah besar *hyper-*

parameter, network ini memiliki *layer convolution* dengan filter 3x3 dan *layer maxpool* dengan filter 2x2. Konfigurasi *layer convolution* dan *maxpool* ini diterapkan secara konsisten di seluruh arsitektur. *Network* ini memiliki 2 *layer fully connected* yang diikuti oleh *softmax* untuk output. Arsitektur VGG-16 memiliki 16 *layer* yang memiliki bobot. *Network* ini adalah jaringan yang cukup besar dan memiliki sekitar 138 juta parameter [5].

B. Transfer learning

Pada penelitian ini digunakan metode *Transfer learning* untuk pembelajaran pada *network* yang memanfaatkan *layer classification*. *Layer classification* dikenal sebagai 3 *layer* terakhir yaitu *layer fully connected*, *softmax*, dan *output*. Tabel 1 menunjukkan nama-nama *layer* yang digunakan pada empat arsitektur *network* yang digunakan pada penelitian ini.

Tabel 1. Classification *layer* pada *network* yang telah dilatih

Network yang telah dilatih	Classification Layer
AlexNet	fc8, prob, output
GoogleNet	loss3-classifier, prob, output
ResNet50	fc1000, fc1000_softmax, classificationlayer_fc1000
VGG16	fc8, prob, output

Layer classification yang semula ini diganti dengan *layer classification* baru yang sesuai dengan tugas klasifikasi baru. *Layer* yang baru ini juga mencakup jumlah kelas baru dan kecepatan pembelajaran yang ditetapkan di *network* yang baru.

Layer classification pada *network* asalnya dikonfigurasi untuk jumlah kelas pada *training* sebelumnya. *Layer* ini pada arsitektur *network* tersebut berisi informasi yang diproses oleh *network* menjadi probabilitas kelas, nilai loss, dan label yang diprediksi. Untuk mengklasifikasikan citra baru, *layer* ini diganti dengan *layer* baru dan disesuaikan dengan dataset yang baru.

Layer fully connected diganti dengan *layer* yang baru yang memiliki jumlah keluaran yang sama dengan jumlah kelas. Untuk membuat pembelajaran lebih cepat pada *layer* yang baru dibandingkan dengan *layer* sebelumnya, faktor kecepatan pembelajaran ditingkatkan.

Layer classification menentukan kelas keluaran *network*. *Layer classification* diganti dengan yang baru tanpa label kelas. Pada saat *training*, *network* secara otomatis menentukan kelas keluaran dari *layer* tersebut.

Network memerlukan citra masukan dengan ukuran tertentu, tetapi citra input memiliki ukuran yang berbeda. Ukuran citra diubah secara otomatis

untuk *training network* tersebut. Selanjutnya dilakukan operasi augmentasi data pada citra *training* dengan menerapkan transformasi geometris affine acak yaitu teknik *reflection* secara horizontal, dan teknik *translation* secara horizontal dan vertikal. Augmentasi data membantu mencegah *network* dari *overfitting* dan mengingat detail yang tepat dari citra *training*.

Pada *training* ditentukan parameter kecepatan pembelajaran untuk *transfer learning* dengan mempertahankan fitur pada *layer* awal *network* yang telah dilatih sebelumnya (bobot *layer* yang ditransfer). Untuk memperlambat pembelajaran di *layer* yang ditransfer, kecepatan pembelajaran awal di-set ke nilai yang kecil. Pada langkah sebelumnya, telah dilakukan peningkatan faktor kecepatan pembelajaran untuk *layer fully-connected* untuk mempercepat pembelajaran di *layer* akhir yang baru. Kombinasi *setting* kecepatan pembelajaran ini menghasilkan pembelajaran yang berjalan cepat hanya di *layer* baru dan pembelajaran lebih lambat di *layer* yang lain.

Saat melakukan *transfer learning*, tidak perlu dilakukan *training* dengan jumlah *epoch* yang sama dengan jumlah sebelumnya. *Epoch* adalah siklus *training* lengkap di seluruh dataset *training*. Pada *training* ini juga ditentukan ukuran *mini batch file* dan data validasi. Validasi *network* dilakukan setiap iterasi dengan frekuensi yang telah ditentukan selama *training*.

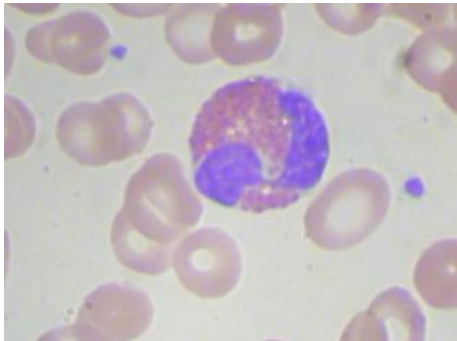
Selanjutnya *training* dilakukan pada *network* yang terdiri dari *layer* yang ditransfer dan *layer* baru dengan menggunakan GPU. Kemudian klasifikasi citra dilakukan dengan menggunakan *network* yang telah melalui *training* tersebut.

C. Dataset dan eksperimen

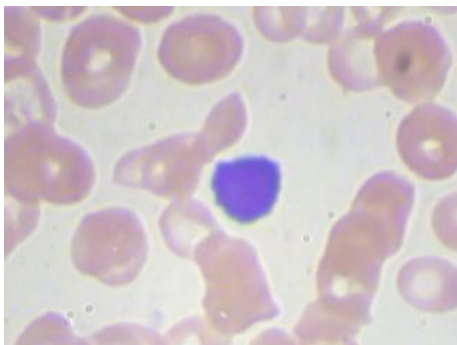
Dataset untuk *training* didapatkan dari Blood Cells Image yang tersedia secara publik pada <https://www.kaggle.com/paultimothymooney/blood-cells>. Pada *training* digunakan 400 citra digital sel darah putih yang terdiri dari 4 kelas yaitu 100 citra *Eosinophil*, 100 citra *Lymphocyte*, 100 citra *Monocyte* dan 100 citra *Neutrophil*. Dengan jumlah dataset untuk *training* sebanyak 400 ini maka proses *training* dapat berlangsung lebih cepat dibandingkan dengan menggunakan jumlah dataset lebih banyak dalam jumlah ribuan. Dengan jumlah dataset tersebut fine-tuning pada *network* yang baru tetap dapat bekerja lebih baik, karena *network* yang baru itu telah belajar untuk mempelajari fitur yang baru tersebut sebagai hasil *training* sebelumnya. Hal ini merupakan kelebihan metode *transfer learning* yang digunakan dalam penelitian ini.

Ukuran tiap citra digital adalah piksel. Pengaturan eksperimental adalah sebagai berikut: dataset dibagi menjadi dua bagian, 70% untuk *training* dan 30% untuk pengujian. Total data untuk tiap kelas adalah

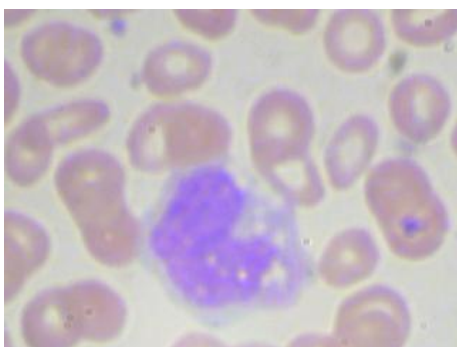
100 citra, 70 citra untuk *training*, dan 30 citra untuk validasi. Ukuran semua citra adalah 320x240 piksel. Untuk keperluan *training* ini dimana diperlukan variasi citra, maka juga digunakan citra yang diaugmentasi. Gambar 2, Gambar 3, Gambar 4 dan Gambar 5 berturut-turut menunjukkan contoh citra *Eosinophil*, *Lymphocyte*, *Monocyte* dan *Neutrophil*.



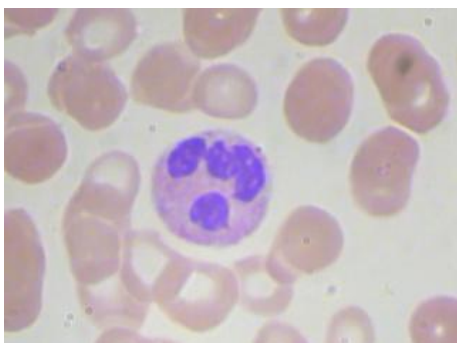
Gambar 2. Citra *Eosinophil*.



Gambar 3. Citra *Lymphocyte*.



Gambar 4. Citra *Monocyte*.



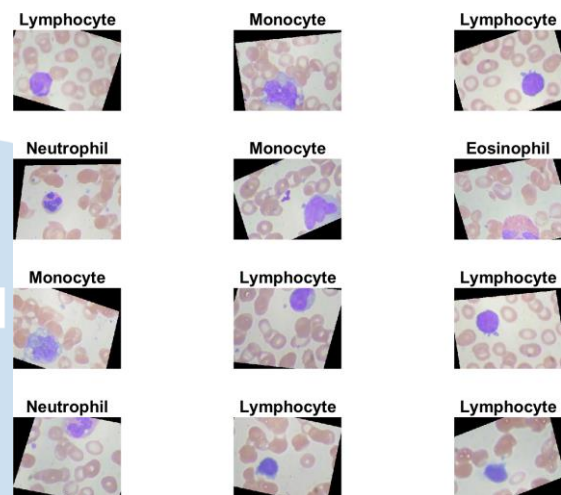
Gambar 5. Citra *Neutrophil*.

Parameter *training* yang digunakan untuk semua *network* adalah sebagai berikut: *learning rate* 1e-4, *minibatch size* 10, *max epoch* 30, *validation frequency* 3.

Arsitektur *network* diimplementasikan dengan MATLAB pada prosesor Intel(R) Core(TM) i7-8565U CPU @ 1.80GHz 1.99 GHz dan RAM 16,0 GB dengan unit pemrosesan grafis (GPU) NVIDIA 1 GB.

IV. HASIL DAN PEMBAHASAN

Pada eksperimen yang telah dilakukan, semua arsitektur *network* berhasil melakukan klasifikasi sel darah putih dengan menggunakan *transfer learning*. Contoh hasil klasifikasi sel darah putih ditunjukkan pada Gambar 6.



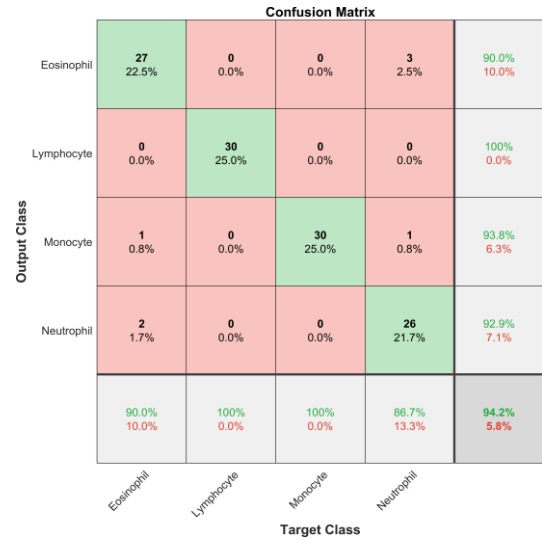
Gambar 6. Contoh hasil klasifikasi sel darah putih.

A. Perbandingan *Confusion matrix*

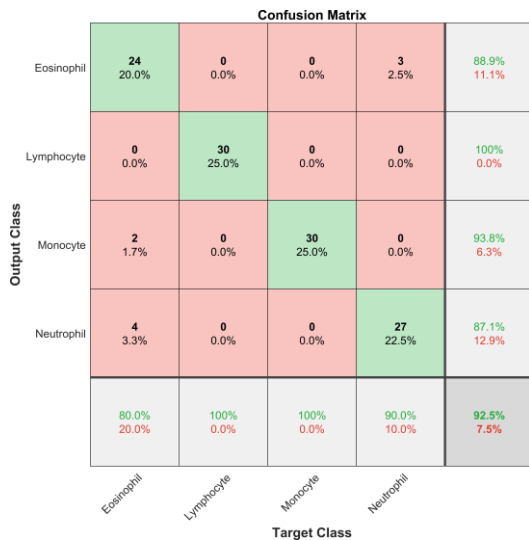
Performa klasifikasi untuk tiap arsitektur *network* disajikan dalam bentuk *Confusion matrix* yang ditunjukkan pada Gambar 7, Gambar 8, Gambar 9 dan Gambar 10. Untuk mengevaluasi hasil klasifikasi tersebut, digunakan empat *performance metrics* yaitu *Accuracy*, *Precision*, *Recall* dan *Fmeasure*. Hasil pengukuran *performance metrics* tersebut disajikan pada Tabel 2.



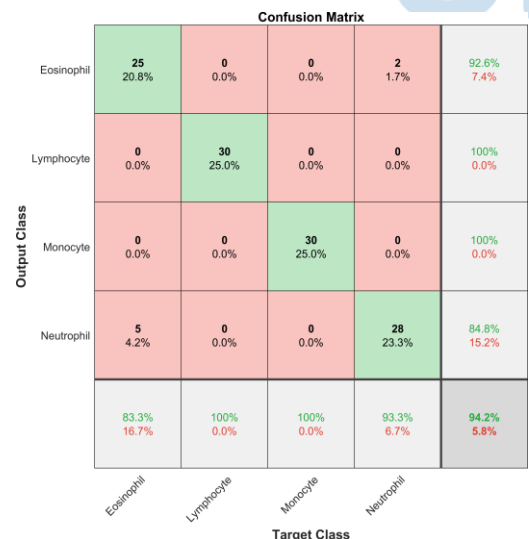
Gambar 7. Confusion matrix AlexNet.



Gambar 10. Confusion matrix VGG-16.



Gambar 8. Confusion matrix GoogleNet.



Gambar 9. Confusion matrix ResNet-50.

Nilai pada diagonal utama CM mewakili semua contoh yang diklasifikasikan dengan benar. Baris di bawah setiap CM menunjukkan tingkat *accuracy* yang dicapai untuk setiap kelas yang diprediksi dan kesalahan prediksi kelas. Gambar 7, Gambar 8, Gambar 9 dan Gambar 10 menunjukkan bahwa jumlah citra yang diklasifikasikan dengan benar pada arsitektur ResNet-50 paling tinggi dibandingkan dengan 3 arsitektur yang lain. Urutan berikutnya adalah arsitektur VGG-16, disusul oleh GoogleNet dan terakhir adalah AlexNet. Urutan yang sama juga terjadi pada perbandingan tingkat *accuracy* yang diukur pada setiap kelas pada kolom paling kanan dan paling bawah. Hal ini menunjukkan bahwa berdasarkan CM tersebut, arsitektur ResNet-50 memiliki performa yang terbaik dalam melakukan klasifikasi tersebut.

B. Evaluasi menggunakan Performance metrics

Tabel 2 menunjukkan *performance metrics* untuk 4 arsitektur *network* yang digunakan. *Performance metrics* yang digunakan adalah *Accuracy*, *Precision*, *Recall* dan *Fmeasure*.

Tabel 2. Performance metrics

	AlexNet	GoogLeNet	Resnet-50	VGG-16
Accuracy	0,9000	0,9333	0,9417	0,9167
Precision	0,8992	0,9348	0,9424	0,9287
Recall	0,9000	0,9333	0,9416	0,9166
Fmeasure	0,8996	0,9340	0,9420	0,9226

Tabel 2 menunjukkan bahwa ResNet-50 menunjukkan performa paling tinggi untuk *metric accuracy* yaitu 94,17% sedangkan GoogleNet, VGG-16, dan AlexNet masing-masing memiliki *accuracy* 93,33%, 91,67% dan 90,00%. Jika perbandingan *accuracy* ini ditinjau dari selisih *accuracy* maka ResNet-50 memiliki *accuracy* dengan selisih 0,84% terhadap GoogleNet, 2,5% terhadap VGG-16 dan 4,17% terhadap AlexNet.

Untuk 3 *metric* yang lain yaitu *Precision*, *Recall* dan *Fmeasure*, Tabel 2 juga menunjukkan urutan nilai *metric* yang sama. ResNet-50 memiliki nilai paling tinggi, disusul oleh GoogleNet, VGG-16 dan AlexNet. Dengan demikian secara keseluruhan berdasarkan CM dan *performance metrics*, urutan performa arsitektur tersebut adalah ResNet-50, GoogleNet, VGG-16, dan AlexNet.

V. SIMPULAN

Dalam penelitian ini dilakukan analisis perbandingan performa arsitektur *network* dalam mengklasifikasikan citra digital sel darah putih. Metode *transfer learning* dengan menggunakan arsitektur *network* yang telah dilatih terlebih dahulu diterapkan untuk mengklasifikasikan citra sel tersebut. Terdapat empat arsitektur *network* yang digunakan yaitu AlexNet, GoogleNet, ResNet-50 dan VGG-16. Untuk meningkatkan jumlah dataset dan mengurangi masalah *overfitting* juga digunakan augmentasi data pada metode yang diterapkan. Untuk mengevaluasi performa tiap arsitektur *network* yang diterapkan maka digunakan CM dan *performance metrics* yaitu *Accuracy*, *Precision*, *Recall* dan *Fmeasure*.

Hasil penelitian ini menunjukkan bahwa empat arsitektur *network* yang digunakan berhasil mengklasifikasikan citra sel darah putih menggunakan metode *transfer learning*. Jumlah citra yang diklasifikasikan dengan benar oleh arsitektur ResNet-50 paling tinggi dibandingkan dengan 3 arsitektur yang lain. Urutan berikutnya adalah arsitektur VGG-16, disusul oleh GoogleNet dan terakhir adalah AlexNet. Untuk *metric accuracy*, ResNet50 mencapai nilai paling tinggi yaitu 94,17% dan disusul oleh arsitektur GoogleNet, VGG-16, dan AlexNet. *Performance metrics* yang lain yaitu *Precision*, *Recall* dan *Fmeasure* menunjukkan urutan yang sama yaitu ResNet-50, GoogleNet, VGG-16 dan AlexNet. Dengan demikian ResNet-50 adalah arsitektur *network* yang menunjukkan performa paling tinggi dalam mengklasifikasikan citra sel darah putih.

Untuk arah penelitian di masa mendatang, disarankan meningkatkan *accuracy* dan nilai *performance metrics* yang lain menggunakan arsitektur *network* dengan menerapkan teknik atau algoritma optimasi.

UCAPAN TERIMA KASIH

Penulis menyampaikan terimakasih kepada Program Studi Informatika Universitas Ciputra Surabaya yang telah memberikan dukungan kepada penulis dalam melakukan penelitian ini.

DAFTAR PUSTAKA

- [1] A. T. Sahlol, P. Kollmannsberger, and A. A. Ewees, "Efficient Classification of White Blood Cell Leukemia with Improved Swarm Optimization of Deep Features," *Sci. Rep.*, vol. 10, no. 1, pp. 1–11, 2020, doi: 10.1038/s41598-020-59215-9.
- [2] Y. Lecun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015, doi: 10.1038/nature14539.
- [3] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Z. Med. Phys.*, vol. 29, no. 2, pp. 102–127, 2019, doi: 10.1016/j.zemedi.2018.11.002.
- [4] L. D. Nguyen, D. Lin, Z. Lin, and J. Cao, "Deep CNNs for microscopic image classification by exploiting transfer learning and feature concatenation," *Proc. - IEEE Int. Symp. Circuits Syst.*, vol. 2018-May, pp. 3–7, 2018, doi: 10.1109/ISCAS.2018.8351550.
- [5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–14, 2015.
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual learning for image recognition," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 2016-December, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.
- [7] C. Szegedy et al., "Going deeper with convolutions," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 07-12-June-2015, pp. 1–9, 2015, doi: 10.1109/CVPR.2015.7298594.
- [8] O. Russakovsky et al., "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015, doi: 10.1007/s11263-015-0816-y.
- [9] H. C. Shin et al., "Deep Convolutional Neural Networks for Computer-Aided Detection: CNN Architectures, Dataset Characteristics and Transfer learning," *IEEE Trans. Med. Imaging*, vol. 35, no. 5, pp. 1285–1298, 2016, doi: 10.1109/TMI.2016.2528162.
- [10] T. Pansombut, S. Wikaisuksakul, K. Khongkraphan, and A. Phon-On, "Convolutional neural networks for recognition of lymphoblast cell images," *Comput. Intell. Neurosci.*, vol. 2019, 2019, doi: 10.1155/2019/7519603.
- [11] A. Çınar and S. A. Tuncer, "Classification of lymphocytes, monocytes, eosinophils, and neutrophils on white blood cells using hybrid Alexnet-GoogleNet-SVM," *SN Appl. Sci.*, vol. 3, no. 4, pp. 1–11, 2021, doi: 10.1007/s42452-021-04485-9.
- [12] A. Kensert, P. J. Harrison, and O. Spjuth, "Transfer learning with Deep Convolutional Neural Networks for Classifying Cellular Morphological Changes," *SLAS Discov.*, vol. 24, no. 4, pp. 466–475, 2019, doi: 10.1177/2472555218818756.
- [13] H. M. Ahmad, S. Ghuffar, and K. Khurshid, "Classification of Breast Cancer Histology Images Using Transfer learning," *Proc. 2019 16th Int. Bhurban Conf. Appl. Sci. Technol. IBCAST 2019*, pp. 328–332, 2019, doi: 10.1109/IBCAST.2019.8667221.
- [14] S. Hosseinzadeh Kassani and P. Hosseinzadeh Kassani, "A comparative study of deep learning architectures on melanoma detection," *Tissue Cell*, vol. 58, no. April, pp. 76–83, 2019, doi: 10.1016/j.tice.2019.04.009.
- [15] V. Maeda-Gutiérrez et al., "Comparison of convolutional neural network architectures for classification of tomato plant diseases," *Appl. Sci.*, vol. 10, no. 4, 2020, doi: 10.3390/app10041245.