Pembuatan Aplikasi *Mobile Augmented Reality* dengan *Scene Recognition* Memanfaatkan *Convolutional Neural Network*

Joseph Nathanael Witanto[1], Gregorius Satia Budhi[2], Liliana[3]
Program Studi Teknik Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121-131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) -8417658
m26413016@john.petra.ac.id[1], greg@petra.ac.id[2], lilian@petra.ac.id[3]

ABSTRAK

Perkembangan teknologi *smartphone* dan kebutuhan masyarakat mencari informasi suatu tempat membuka kesempatan untuk pemanfaatan *augmented reality* untuk membantu memperoleh informasi tempat sambil melihat melalui kamera. Sistem AR berbasis sensor mengandalkan GPS yang tidak bisa selalu tersedia. Pada penelitian ini digunakan sistem AR dengan memanfaatkan *convolutional neural network* untuk mendeteksi *scene*.

Terdapat 45 variasi *network* yang akan diuji untuk menemukan kombinasi yang terbaik. Parameter yang dibedakan adalah arsitektur, metode inisialisasi, dan fungsi aktivasi yang digunakan. Arsitektur yang digunakan adalah GoogLeNet dan 2 variasi GoogLeNet yang diperkecil. Metode inisialisasi yang digunakan adalah inisialisasi secara *random* (Xavier dan MSRA) dan inisialisasi menggunakan *weight* yang sudah di-*training* sebelumnya. Fungsi aktivasi yang digunakan adalah ReLU, PReLU, dan ELU. Augmentasi data yang dilakukan saat *training* berupa *random cropping*, *color balance*, rotasi, *blur*, *sharpen*, serta manipulasi *brightness* dan *contrast*.

Dari 1649 foto di 12 kategori scene, digunakan 321 foto untuk *testing* dengan variasi rotasi (interval 30 derajat), *blur*, *sharpen*, *brightness*, dan *contrast*. Didapati bahwa jaringan dengan metode inisialisasi dengan *finetuning* di seluruh bagian *network* dan fungsi aktivasi PReLU memiliki rata-rata akurasi yang lebih baik.

Kata Kunci: *Convolutional Neural Network*, *Scene Recognition*, *Mobile Augmented Reality*, Arsitektur *Neural Network*.

ABSTRACT

The development of smartphone technology and the need of people to gather information of places open chance to utilize augmented reality to help people get the information of a scene and see the scene through camera at the same time. The sensor-based AR system depends on GPS which is not always available. This research uses AR system using convolutional neural network for scene recognition.

There are 45 network variations that will be tested to find the best combination. The different parameters that will be used are architecture, initialization method, and activation function that will be used. The architectures used are GoogLeNet and 2 variations of simplified GoogLeNet. The initialization methods used are random-based (Xavier and MSRA) and pretrained weights. The activation functions used are ReLU, PReLU, and ELU. The data augmentations used during training are random cropping, color balance, rotation, blur, sharpen, and brightness-contrast manipulation.

Out of 1649 photos from 12 scene categories, 321 photos will be used for testing with variations on rotation (30 degrees interval), blur, sharpen, brightness, and contrast. Network with initialization method using finetuning on all areas of network and PReLU activation function has better average accuration.

Keywords: *Convolutional Neural Network, Scene Recognition, Mobile Augmented Reality, Neural Network Architecture.*

1. PENDAHULUAN

Pada era teknologi ini, konsep tentang *smart city* mulai berkembang, dimana teknologi informasi diintegrasikan ke dalam suatu kota. Salah satu kebutuhan masyarakat adalah untuk menjelajahi suatu kota dan mencari informasi suatu tempat berdasarkan foto tempat tersebut. Pendatang dari luar kota yang baru berkunjung dan mengalami kesulitan dapat difasilitasi dengan informasi tempat yang ada.

Teknologi *augmented reality* (AR) pada *mobile device* dapat digunakan untuk membantu masyarakat memperoleh informasi suatu tempat sambil melihat kondisi sekitar melalui kamera. Sistem AR pada *mobile device* dapat memanfaatkan data posisi pengguna (GPS) dan orientasi dari kamera. Namun, kelemahan dari sistem ini adalah kebergantungan sepenuhnya pada koneksi jaringan dan kualitas sensor orientasi dari kamera. Untuk menjawab masalah ini, sistem AR dapat memanfaatkan *computer vision*, yaitu melalui pengenalan suatu tempat dari foto (*scene recognition*). Sistem ini dapat digunakan pada kondisi tanpa jaringan sehingga pengguna dapat memperoleh data tempat yang diinginkan hanya melalui referensi foto tempat.

Penelitian ini berguna untuk membuat aplikasi yang memanfaatkan teknologi *augmented reality* dan *scene recognition* sehingga informasi mengenai sebuah foto lokasi dapat disampaikan secara interaktif.

2. LANDASAN TEORI

2.1 Augmented Reality

Augmented Reality (AR) adalah teknologi yang menciptakan versi dari realita yang dimodifikasi, diaugmentasi dengan informasi digital (virtual) pada layar komputer atau perangkat *mobile*.

Sistem AR berbasis sensor memanfaatkan sensor lokasi dan sensor orientasi dari perangkat *mobile* untuk memperoleh lokasi pengguna di dunia nyata. Sistem AR berbasis *computer vision* memanfaatkan data dari kamera untuk diproses lebih lanjut menggunakan algoritma *image processing* dan *computer vision* untuk menganalisa dan mendeteksi objek [4].

2.2 Convolutional Neural Network

Algoritma *learning* adalah algoritma kecerdasan buatan yang mampu belajar dari data. Salah satu metode yang digunakan adalah jaringan saraf tiruan (*neural network*), yaitu jaringan yang terdiri dari banyak unit yang melakukan fungsi nonlinear untuk memetakan data menjadi hasil tertentu [8]. *Convolutional Neural Network* (CNN) adalah tipe *neural network* dimana struktur jaringannya mengikuti konsep dari proses *convolution*, dimana konektivitas jaringan berada di *region square* tertentu dan *weight* dari jaringan digunakan seperti filter dalam proses *convolution* dan digunakan bersama untuk seluruh *region square* dari input (*shared weight*).

2.2.1 Arsitektur

Szegedy, et al. (2015) mengadakan penelitian untuk arsitektur *neural network* untuk klasifikasi objek pada kompetisi ImageNet Large-Scale Visual Recognition Challenge 2014 (ILSVRC14). Arsitektur ini diberi nama Inception dan didesain dengan mempertimbangkan *budget* untuk melakukan komputasi dan mempertimbangkan faktor efisiensi untuk penggunaan pada perangkat *mobile*. Perkembangan dari arsitektur ini dan dikumpulkan untuk kompetisi adalah GoogLeNet, jaringan dengan kedalaman 22 *layer*. Dengan jumlah parameter 12 kali lebih sedikit dibandingkan dengan penelitian Krizhevsky, Sutskever, & Hinton (2012), GoogLeNet memiliki tingkat akurasi lebih tinggi (top-5% *error* 6.67% dibandingkan dengan top-5% *error* 15.3%) [10].

2.2.2 Fungsi Aktivasi

He, Zhang, Ren, & Sun (2015) mengembangkan fungsi aktivasi Parametric Rectified Linear Unit (PReLU) untuk menggantikan ReLU. PReLU dapat meningkatkan kemampuan model untuk melakukan transformasi dengan sedikit komputasi ekstra dan risiko *overfitting* yang kecil. Fungsi pada PReLu adalah max (x, 0) + α * min(0, x) dimana a adalah variabel yang dapat dilatih. Jika α = 0, PReLU pada dasarnya melakukan transformasi yang sama dengan ReLU, untuk α = 1, PReLU akan melakukan transformasi linear, dan untuk α mendekati 0 PReLU akan sama dengan Leaky ReLU [5].

Clevert, Unterthiner, Hochreiter (2016) mengembangkan Exponential Linear Unit (ELU) untuk proses *learning* yang lebih cepat dan akurat di *deep neural network*. ELU memiliki nilai negatif, sehingga memampukan jaringan memiliki nilai aktivasi rata-rata mendekati 0. Properti inilah yang menjadi alasan suksesnya LReLU dan PReLU. Berbeda dengan LReLU dan PReLU, ELU memiliki wilayah saturasi yang jelas di daerah negatif, sehingga yang dipelajari memiliki representasi yang lebih stabil. Hasil eksperimen menunjukkan bahwa ELU memiliki hasil yang lebih akurat dibandingkan dengan fungsi aktivasi lain di beberapa *dataset* dengan optimasi jaringan menggunakan *Stochastic Gradient Descent* [3].

2.2.3 Inisialisasi Weight

Agrawal, Girshick, & Malik (2014) meneliti tentang efek *transfer learning* pada *deep neural network*. Mereka mendapati bahwa jaringan dengan data *training* yang cukup banyak tetap menerima dampak positif dari *pre-training* (dalam penelitian ini menggunakan ImageNet *dataset*) dan *finetuning* (melakukan adaptasi untuk *dataset* target). Hal lain yang mereka temukan adalah proses *transfer learning* ini tidak menyebabkan *overfitting* (*error* disebabkan karena jaringan tidak dapat melakukan generalisasi, yaitu kemampuan mengenali objek di luar data *training* yang disediakan) [1].

2.2.4 Augmentasi Data

Chatfield, Simonyan, Vedaldi, & Zisserman (2014) melakukan eksperimen dan mendapati bahwa augmentasi data secara konsisten meningkatkan performa jaringan sekitar 3%. Mereka melakukan mengubah ukuran gambar sehingga dimensi paling kecil berukuran 256 piksel. Kemudian diambil potongan gambar 224 x 224 dari 4 pojok dan tengah gambar. Selain itu dilakukan *flip* secara horizontal untuk melatih invarian terhadap refleksi [2].

Howard (2014) mengadakan penelitian dengan augmentasi data untuk mengembangkan jaringan yang invarian terhadap translasi dan warna. Ukuran gambar diubah dengan sisi yang kecil berukuran 256, kemudian dilakukan *random crop* untuk data *training* berukuran 224x224. *Random crop* lebih meningkatkan invarian terhadap translasi dibandingkan di pojok dan tengah gambar saja. Setelah itu, dilakukan manipulasi kontras, kecerahan (*brightness*) dan warna menggunakan *python image library* [6].

Ronneberger, Fischer, & Brox (2015) mengadakan penelitian tentang segmentasi gambar biomedis dan hanya sedikit data *training* yang tersedia. Mereka mengaplikasikan rotasi dan deformasi gambar sehingga jaringan dapat belajar invarian terhadap rotasi dan deformasi tanpa perlu melihat adanya transformasi tersebut dalam data *training*. Hal tersebut penting dalam segmentasi biomedis karena deformasi sel sering terjadi [7].

2.2.5 Dropout

Deep neural network dengan jumlah parameter yang besar memiliki kemampuan yang besar namun memiliki masalah yang serius dengan *overfitting*. Selain itu jaringan yang lebih besar juga lebih lambat digunakan, sehingga mengatasi masalah *overfitting* dengan mengkombinasikan prediksi beberapa jaringan yang besar memakan waktu yang lama dan *resource* yang besar.

Dropout adalah teknik untuk mengatasi masalah ini dengan secara acak meniadakan unit selama proses *training*. Hal ini mengurangi kemungkinan unit yang ada tidak fokus ke pola tertentu yang sebenarnya merupakan *noise*. Srivastava, Hinton, Krizhevsky, Sutskever, Salakhutdinov (2014) menemukan bahwa *dropout* meningkatkan performa *neural network* di bidang *vision*, *speech recognition*, *computational biology*, dan klasifikasi dokumen. Kekurangan *dropout* adalah meningkatnya waktu *training* (2-3 kali lebih lama dibandingkan dengan *neural network* dengan arsitektur yang sama) [9].

3. DESAIN SISTEM

3.1 Variasi Jaringan

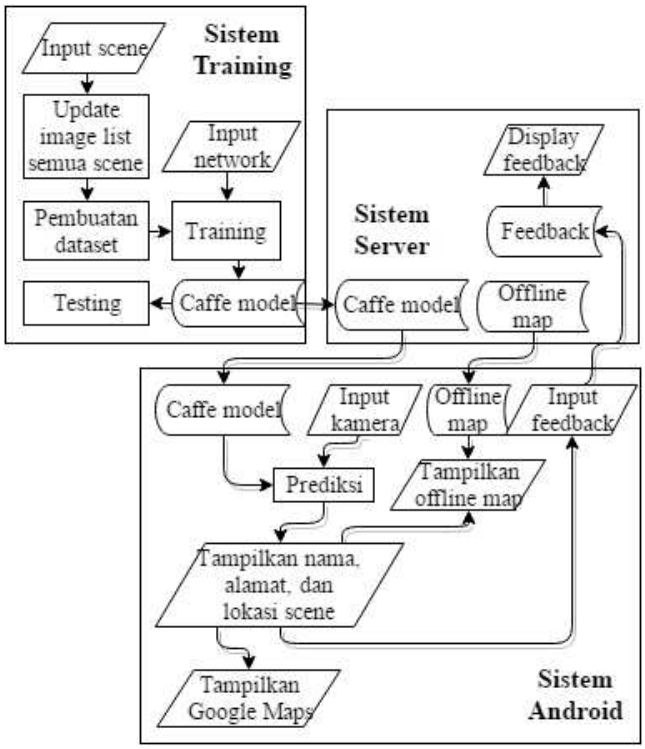
Terdapat 45 variasi jaringan yang akan diuji untuk menemukan kombinasi yang terbaik. Parameter yang dibedakan adalah arsitektur, metode inisialisasi, dan fungsi aktivasi yang digunakan.

Arsitektur yang digunakan adalah GoogLeNet dan 2 variasi GoogLeNet yang diperkecil. Metode inisialisasi yang digunakan adalah inisialisasi secara *random* (Xavier dan MSRA) dan

inisialisasi menggunakan *weight* yang sudah di-*training* sebelumnya. Fungsi aktivasi yang digunakan adalah ReLU, PReLU, dan ELU.

3.2 Sistem Aplikasi

Sistem aplikasi pada tugas akhir ini dibagi menjadi tiga sistem, yaitu sistem *training*, *server*, dan Android seperti ditunjukkan melalui *Block Diagram* pada Gambar 1.



Gambar 1. *Block Diagram* Sistem Aplikasi

Sistem *training* merupakan sistem untuk melakukan proses input data gambar yang kemudian digunakan untuk melatih dan melakukan uji coba terhadap jaringan. Pada fase awal dilakukan percobaan terhadap beberapa variasi struktur jaringan hingga didapatkan struktur yang optimal.

Sistem *server* merupakan sistem untuk menyimpan model yang telah dihasilkan dari proses *training* dan juga *offline map* untuk dapat diunduh oleh Android. Sistem ini juga menerima *feedback* dari sistem Android apabila pengguna ingin mengirimkan label tertentu untuk gambar yang diambil menggunakan kamera.

Sistem Android merupakan sistem yang digunakan oleh pengguna untuk mengambil gambar dengan kamera dan memperoleh deskripsi dari hasil klasifikasi *scene*. Pengguna dapat melihat *scene* tersebut pada *offline map*. Selain itu, pengguna dengan koneksi internet dapat mengirimkan *feedback* ke *server* ataupun memperoleh data tambahan dari Google Maps.

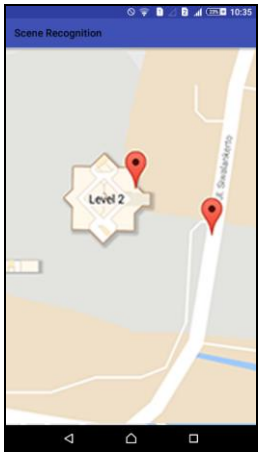
4. PENGUJIAN SISTEM

4.1 Aplikasi Android

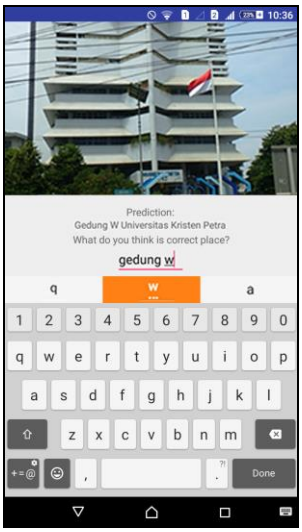
Aplikasi *mobile* yang telah dibuat akan melakukan recognition terhadap *scene* dan menampilkan nama, alamat, dan jarak antara posisi *scene* dan pengguna seperti terlihat pada Gambar 2. Pengguna juga dapat membuka peta *offline* (Gambar 3), Google Maps (Gambar 4), dan mengirim saran untuk kategori tempat yang difoto oleh pengguna (Gambar 5).

Gambar 2. Tampilan Halaman Capture

Gambar 3. Tampilan Halaman Offine Map



Gambar 4. Tampilan Halaman Google Maps



Gambar 5. Tampilan Halaman Send Feedback

4.2 Proses *Training*

Program akan diuji dengan melakukan proses *training* dan *testing* terhadap berbagai variasi jaringan menggunakan 12 kategori *scene*. Setelah itu akan dilakukan proses *test* pada jaringan berupa akurasi rata-rata, perubahan akurasi yang disebabkan oleh perubahan citra (rotasi, *blur*, *sharpen*, *brightness*, dan *contrast*), ukuran *file* untuk menyimpan *weight*, dan waktu yang dibutuhkan oleh jaringan untuk melakukan klasifikasi pada *smartphone*.

Adapun jaringan pada penelitian ini digunakan caffemodel (*file* untuk menyimpan *weight* jaringan di *library* Caffe) yang sudah dilatih menggunakan *database* Places yang berisi 205 kategori tempat [11].

Training dilakukan menggunakan *library* Caffe sebanyak 4000 iterasi untuk tiap jaringan. Augmentasi data yang dilakukan saat *training* berupa *random cropping*, *color manipulation*, rotasi, *blur*, *sharpen*, serta manipulasi *brightness* dan *contrast*. Fungsi untuk mengoptimasi jaringan menggunakan Adam (*adaptive moment estimation*) pada *library* Caffe. Dari 1649 gambar, digunakan 1328 untuk *training*. Dalam proses *training*, digunakan *batch* sebesar 50 gambar dalam 1x iterasi dan akurasi dihitung dengan jumlah gambar yang berhasil dideteksi dibagi dengan 50.

Jaringan yang dilatih menggunakan metode *finetuning* seluruh bagian *network* cenderung mengalami peningkatan akurasi dengan lebih cepat. Metode *finetuning* di bagian akhir *network* mengalami peningkatan akurasi yang kurang stabil (perubahan akurasi berupa naik dan turun memiliki rentang yang cukup besar) dibandingkan dengan jaringan yang menggunakan metode inisialisasi *random* (Xavier dan MSRA). Hal ini disebabkan karena adanya ketidaksesuaian fitur yang dipelajari di Places Database dengan *dataset* yang digunakan di penelitian ini, sehingga ada bagian yang menghalangi saat klasifikasi dan tidak dapat dirubah (sebagian *weight* di Finetuning B dan C sengaja tetap menggunakan *weight* yang dilatih menggunakan Places Database).

Didapati bahwa jaringan dengan fungsi aktivasi PReLU cenderung lebih cepat dan stabil dalam peningkatan akurasi, meskipun metode inisialisasinya menggunakan *finetuning* di bagian akhir *network*. Jaringan dengan fungsi aktivasi ELU cenderung tidak stabil (rentang perubahan akurasi lebih besar) dibandingkan ReLU.

4.3 Proses *Testing*

Proses ini dilakukan setelah selesai dilakukan *training* terhadap jaringan. *Testing* akan dilakukan terhadap 321 dari 1649 gambar. Untuk setiap gambar akan dilakukan augmentasi dengan menggunakan rotasi, *blur* (3 variasi), *contrast* (3 variasi), dan *brightness* (3 variasi).

Didapati bahwa 3 jaringan dengan akurasi terbaik diperoleh oleh jaringan dengan arsitektur SmallNet, fungsi PReLU atau ReLU, dan metode inisialisasi Finetuning A (seluruh bagian *network*), MSRA, atau Xavier. Peringkat 4 dan 5 terbaik adalah jaringan MediumNet dengan fungsi PReLU atau RELU dan metode inisialisasi Xavier. 3 jaringan dengan akurasi terburuk juga memiliki arsitektur SmallNet dan metode inisialisasi Finetuning A, MSRA, atau Xavier. Perbedaannya adalah pada fungsi yang digunakan, yaitu ELU. Peringkat 4 dan 5 terburuk merupakan jaringan dengan arsitektur GoogLeNet dengan fungsi ReLU atau ELU dan metode inisialisasi *finetuning* di bagian akhir *network* saja.

Didapati bahwa fungsi aktivasi memiliki peran penting dalam penentuan akurasi, ditandai dengan selisih rata-rata akurasi sebesar 10.96% antara ELU (rata-rata akurasi paling rendah) dan PReLU (rata-rata akurasi paling tinggi).

Tabel 1. Rata-rata akurasi untuk setiap fungsi aktivasi

Fungsi Aktivasi	Rata-rata akurasi
ReLU	80.44%
PReLU	81.37%
ELU	70.41%

Pada Tabel 2, dilakukan perhitungan akurasi setiap arsitektur untuk jaringan dengan dan tanpa fungsi ELU karena jaringan dengan ELU mengakibatkan perubahan ekstrem terhadap akurasi. GoogLeNet dan MediumNet lebih stabil terhadap fungsi aktivasi (perubahan 0.28% dan 1.16%). SmallNet rentan terhadap perubahan fungsi aktivasi (perubahan 8.77%). Hal ini membuktikan kombinasi arsitektur dan fungsi aktivasi mempengaruhi.

Tabel 2. Rata-rata akurasi untuk setiap fungsi aktivasi

Fungsi Aktivasi	Rata-rata akurasi (dengan ELU)	Rata-rata akurasi (tanpa ELU)	Perubahan
ReLU	78.61%	78.88%	0.28%
PReLU	80.73%	81.89%	1.16%
ELU	73.16%	81.93%	8.77%

Metode *finetuning* lebih stabil terhadap perubahan fungsi aktivasi dibandingkan metode inisialisasi *random*. Xavier dan MSRA rentan terhadap perubahan fungsi aktivasi (perbedaan 5.47% dan 6.48%).

Dilakukan perhitungan akurasi berdasarkan augmentasi yang dilakukan pada *image* yang digunakan untuk *training* maupun *testing*. Didapati bahwa rotasi lebih cenderung berpengaruh

terhadap akurasi, ditandai dengan standar deviasi 2.97.%, lebih tinggi daripada tipe *blur* (standar deviasi 0.853%), tipe *brightness* (standar deviasi 1.19%) dan tipe *contrast* (1.02%).

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan hasil pengujian, dapat disimpulkan beberapa hal:

- Metode inisialisasi finetuning seluruh bagian *network* adalah pilihan yang baik untuk mencapai akurasi tertinggi dan rata-rata akurasi yang paling tinggi.
- Jaringan dengan arsitektur MediumNet menghasilkan akurasi yang baik terlepas dari pilihan fungsi aktivasi, sedangkan SmallNet menghasilkan rata-rata akurasi terbaik jika menggunakan fungsi aktivasi selain ELU. Jaringan dengan arsitektur SmallNet membutuhkan waktu paling kecil. Jaringan dengan fungsi aktivasi PReLU memiliki rata-rata akurasi terbaik, hanya menambah sedikit ukuran *file weight*, dan lebih cepat dibandingkan fungsi aktivasi ELU.
- Manipulasi yang paling berpengaruh terhadap akurasi jaringan adalah rotasi, diikuti dengan *brightness*, *contrast*, dan *blur*.
- Jaringan dengan arsitektur SmallNet, fungsi aktivasi PReLU, dan metode inisialisasi *finetuning* untuk seluruh bagian *network* memiliki peringkat yang baik untuk kriteria rata-rata akurasi, waktu, ukuran file weight, standar deviasi rotasi, selisih *blur*, selisih *brightness*, dan selisih *contrast*

5.2 Saran

- Menambah data gambar untuk *training* dan *testing* dengan menggunakan data dari video yang diambil tiap interval *frame* tertentu.
- Menambah augmentasi data dengan menambahkan *noise* pada gambar yang digunakan untuk *training* dan *testing*.

6. DAFTAR PUSTAKA

[1] Agrawal, P., Girshick, R., & Malik, J. 2014. Analyzing the Performance of Multilayer Neural Networks for Object Recognition. *European Conference on Computer Vision*, (ss. 329-344).

[2] Chatfield, K., Simonyan, K., Vedaldi, A., & Zisserman, A. 2014. Return of the Devil in the Details: Delving Deep into Convolutional Nets. *British Machine Vision Conference*, (ss. 1-12).

[3] Clevert, D.-A., Unterthiner, T., & Hochreiter, S. 2016. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). *International Conference on Learning Representations.*

[4] Grubert, J., & Grasset, R. 2013. *Augmented Reality for Android Application Development*. Birmingham: Packt Publishing.

[5] He, K., Zhang, X., Ren, S., & Sun, J. 2015. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *IEEE International Conference on Computer Vision*, (ss. 1026-1034).

[6] Howard, A. G. 2014. Some Improvements on Deep Convolutional Neural Network Based Image Classification. *International Conference on Learning Representations.*

[7] Ronneberger, O., Fischer, P., & Brox, T. 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Medical Image Computing and Computer-Assisted Intervention.*

[8] Russel, S. J., & Norvig, P. 2010. *Artificial Intelligence*. Upple Saddle River: Prentice Hall.

[9] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 1929-1958.

[10] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., . . . Rabinovich, A. 2015. Going Deeper with Convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, (ss. 1-9).

[11] Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., & Oliva, A. 2014. Learning Deep Features for Scene Recognition using Places Database. *Advances in Neural Information Processing Systems*.