PENCARIAN OBJEK WISATA BERSEJARAH DI PULAU JAWA MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK

Nadia Ramadhani [1] Janson Hendryli [2] Dyah Erny Herwindianti [3]
[1] Teknik Informatika, FTI, Universitas Tarumanaraga
Jl. Letjen S Parman no 1, Jakarta 11440 Indonesia
[1] ramadhaninadia@yahoo.com
[2] jansonh@fti.untar.ac.id
[3] dyahh@fti.untar.ac.id

ABSTRACT

In order to make a image-based search for historical sites in Java, Convolutional Neural Network is used. Not only to make a website which can search historycal sites in Java using image, it also compared two CNN structure. There are two architechture which is used in this paper which is Residual Network and Inception. There are some experiments that were done to establish the best architecture among the two for this aplication. Those experiments showed that inception gave a better result for the application,

Key words

Convolutional Neural Network, Image Searching, Inception, Residual Network

1. Pendahuluan

Objek wisata merupakan suatu hal yang selalu menarik perhatian masyarakat luas. Perhatian tersebut membawa masyarakat untuk mencari tempat atau objek wisata yang memenuhi kriteria mereka dimana lingkup objek wisata tersebut sangatlah beragam.

Indonesia memiliki beragam objek wisata, namun keterbatasan pengetahuan mengenai luasnya objek wisata bersejarah di pulau Jawa membuat para wisatawan menjadi kebingungan dan akhirnya mengunjungi objek wisata yang sama dan pernah mereka kunjungi sebelumnya.

Pencarian di search engine pun dapat dilakukan, namun search engine umumnya hanya dapat mencari menggunakan kata-kata atau kata kunci sehingga jika seorang wisatawan hanya memiliki gambar dan tidak mengetahui nama dari tempat tersebut tidak dapat menemukannya dengan mudah. Sebagai contoh jika seseorang menemukan sebuah gambar objek wisata yang menarik di sebuah majalah dan ingin mengunjunginya, akan sulit bagi orang tersebut untuk mencari nama dari tempat yang berada di gambar tersebut.

Oleh karena itu, untuk mengatasi permasalah ini, dibuat sebuah sistem yang diimplementasikan dalam bentuk website. Situs web ini akan ditujukan untuk para wisatawan agar dapat mencari objek wisata dengan input gambar serta output yang merupakan nama, lokasi dan informasi singkat dari gambar yang dimasukan. Selain itu, situs ini dapat memberikan beberapa lokasi objek wisata bersejarah yang serupa dengan input beserta dengan berbagai informasi ringkas mengenai objek wisata tersebut.

Sistem pencarian Objek Wisata Bersejarah di Pulau Jawa ini dirancang dengan cara menggunakan CNN (Convolutional Neural Network) untuk mengenali objek wisata apa yang dijadikan input. Hasil pengenalan tersebut berupa nama dari objek wisata yang kemudian digunakan untuk mengambil data tambahan dari basis data milik sistem, sehingga menghasilkan lokasi, dan informasi tambahan mengenai objek wisata tersebut, serta akan mengeluarkan sejumlah tempat yang serupa. Pada pembuatan sistem ini akan digunakan Inception v3 dan Residual Network (ResNet) sebagai arsitektur CNN.

Batasan masalah dari perancangan ini adalah pada erancangan website ini menggunakan Bahasa pemograman PHP dan SQL, sedangkan pelatihan model yang digunakan sistem menggunakan bahasa Python. Selanjutnya sistem ini hanya dapat mencari objek wisata bersejarah berupa Candi dan Museum yang berada di pulau Jawa. Batasan terakhir adalah input aplikasi ini merupakan sebuah gambar tampak depan dari objek wisata dengan resolusi minimal 400 x 400 pixels.

Tujuan dari pembuatan Pencarian Objek Wisata Bersejarah di Pulau Jawa Menggunakan convolutional neural network ini adalah untuk merancang sebuah sistem yang dapat menghasilkan nama, lokasi, dan sekilas informasi tentang objek wisata dengan input sebuah gambar tampak depan dari objek wisata. Selain itu juga untuk merancang sebuah sistem yang dapat menghasilkan objek wisata sejenis dengan sesuai dengan pilihan lokasi.

2. Convolutional Neural Network

Aplikasi ini memiliki sebuah fitur utama yaitu fitur pengenalan objek wisata. Cara kerja dari fitur ini adalah pengguna aplikasi memasukan sebuah gambar tampak depan dari objek wisata di pulau Jawa dengan resolusi minimal 400 x 400 pixels. Selanjutnya sistem akan memproses gambar tersebut dan menghasilkan output berupa nama dari objek wisata tersebut. Dengan

informasi nama tersebut, sistem mengambil data berupa lokasi, informasi dan gambar tambahan dari objek wisata tersebut kemudian menampilkan sebagai output juga. Informasi yang disediakan merupakan informasi sejarah singkat dari objek wisata tersebut.

Untuk itu digunakan Convolutional Neural Network. Convolutional Neural Network merupakan bagian dari Neural Network yang memiliki neuron yang memiliki weight, bias dan activation function. Convolutional Neural Network memiliki 3 layer yaitu: Convolutional layer, Pooling Layer, dan Fully-Connected Layer[2].

Convolutional Layer adalah sebuah layer pertama dari CNN, dimana layer ini memiliki sebuah kumpulan filter yang dapat digunakan untuk mempelajari input. Melalui layer ini, fitur akan di ekstraksi dan kemudian di lanjutkan ke layer berikutnya dengan tujuan untuk mengekstraksi fitur yang lebih kompleks[3],

$$(\mathrm{I} * \mathrm{K})_{\mathrm{ij}} = \sum_{m=0}^{k1-1} \square \sum_{n=0}^{k2-1} I(i-m, j-n)K(m,n)$$

.....(1)

Rumus (1) digunakan untuk menghitung matriks filter yang digunakan. Dengan I merupakan matriks *input* dengan ukuran (i x j).

Selanjutnya adalah Pooling layer yang menggabungkan output dari neuron-neuron pada suatu lapisan dengan neuron di lapisan berikutnya. Sebagai contoh, max pooling menggunakan nilai maksimum dari masing-masing kelompok neuron pada lapisan sebelumnya. Contoh lain adalah Average Pooling, yang menggunakan nilai rata-rata dari masing-masing kelompok neuron pada lapisan sebelumnya . Umumnya, metode pooling yang digunakan adalah max pooling atau mengambil nilai terbesar dari bagian tersebut. Dalam sistem yang dibuat, metode yang digunakan adalah max pooling. Hasil dari layer ini adalah multidimensional array, sehingga kita harus melakukan "flatten" atau reshape feature map menjadi sebuah vector agar bisa kita gunakan sebagai input dari fully-connected layer.

Fuly-connected layer memiliki sebagian besar parameter, sehingga lapisan ini rentan terhadap overfitting. Salah satu metode untuk mengurangi overfitting adalah dropout. Pada setiap tahap pelatihan, setiap node dapat di "dropout" dari jaringan atau disimpan, sehingga jaringan yang dihasilkan adalah jaringan yang telah dikurangi. Hanya jaringan yang dikurangi yang dilatih pada data dalam tahap itu. Node yang dihapus kemudian dimasukkan kembali ke jaringan dengan bobot aslinya.

Pada pembuatan sistem ini dibandingkan dua arsitektur Convolutional Neural Network yaitu Inception dan Residual Network. Kemudian dipilih dengan hasil akurasi yang paling besar untuk digunakan di aplikasi akhirnya.

2.1 Inception

Inception v3 adalah sebuah model deep convolutional network yang dikembangkan oleh Google memenuhi ImageNet Large Visual Recognition Challenge pada tahun 2012. Model inception menggunakan beberapa filter pada layer convolutional, tidak seperti convolutional layer yang biasa. Hasil dari beberapa filter tersebut dijadikan satu lagi menggunakan Channel Concat sebelum masuk kedalam iterasi berikutnya.[4]

Tujuan dari modul inception adalah untuk bertindak sebagai multi-level feature extractor dengan menghitung filter-filter convolutions dalam modul yang sama. Hasil dari filter-filter tersebut kemudian ditumpukan kedalam dimensi channel sebelum dimasukan kedalam lapisan selanjutnya.

Dalam pembuatan aplikasi ini seluruh percobaan menggunakan tiga filter yaitu 1x1, 3x3, dan 5x5.

2.2 Residual Network (ResNet)

ResNet (Residual Network) adalah sebuah model arsitektur dengan "skip connection" dan normalisasi heavy batch. Perbedaan terbesar antara Resudial Network dengan lapisan convolution biasa adalah bahwa ResNet memiliki koneksi pintas (skip connection) dengan convolution layer normalnya.[3]

ResNet menggunakan koneksi pintas untuk menghindari masalah gradient yang menghilang, dengan menggunakan kembali aktivasi dari lapisan sebelumnya sampai lapisan selanjutnya yang sekarang mempelajari bobotnya. Selama pelatihan, bobot akan beradaptasi untuk memperkuat lapisan di samping arus. Dalam kasus yang paling sederhana, hanya bobot untuk koneksi selanjutnya disesuaikan, tanpa bobot eksplisit untuk lapisan sebelumnya. Ini biasanya berfungsi dengan baik ketika lapisan non-linear tunggal dilangkahi, atau dalam kasus ketika lapisan menengah semuanya linear. Jika tidak, maka matriks bobot eksplisit harus dipelajari untuk koneksi yang dilewati.

2.3 Backpropagation

Backpropagations adalah metode yang digunakan dalam Artificial Neural Network untuk menghitung nilai gradient yang diperlukan dalam proses penghitungan bobot yang akan digunakan dalam jaringan. Backpropagation biasanya digunakan untuk melatih deep neural network.

Algoritma yang digunakan adalah sebagai berikut:

1. Setiap unit input (Xi) akan menerima sinyal input dan akan menyebarkan sinyal tersebut pada tiap hidden unit (Zj). Setiap hidden unit kemudian akan menghitung aktivasinya dan mengirim sinyal (z) ke tiap unit output. Kemudian setiap unit output (Yk) juga akan menghitung aktivasinya (yk) untuk menghasilkan respons terhadap input yang diberikan jaringan.

$$Zin_k = w0 + \sum_{n=1}^{n} Xt\,Vtj \qquad (2)$$

2. Kemudian memakai fungsi aktivasi yang telah ditentukan untuk menghitung sinyal *output* dari unit *output* yang bersangkutan:

$$Y_k = \frac{1}{1 + e^{-yin_k}} \quad \ldots\ldots\ldots\ldots\ldots\ldots \qquad (3)$$

3. Setiap unit *output* menerima suatu target yang akan dibandingkan dengan *output* yang dihasilkan. Faktor δk ini digunakan untuk menghitung koreksi error (Δwjk) di mana:

$$\Delta w_{jk} = \alpha \delta_k z_j \ldots\ldots\ldots\ldots \qquad (4)$$

4. Setiap *hidden unit* (Zj,j=1,…,p) menjumlah *input* delta (yang dikirim dari lapisan pada langkah sebelumnya yang sudah berbobot.

$$(\delta in_j) = \sum_{k=1}^{m} \delta_k W_{jk} \quad \ldots\ldots\ldots\ldots\ldots\ldots \qquad (5)$$

5. Kemudian hasilnya dikalikan dengan turunan dari fungsi aktivasi yang digunakan jaringan untuk menghasilkan faktor koreksi eror δj, dimana:

$$\delta_j = \delta\_i_{nj}\, f'(z\_i_{nj}) \ldots\ldots \qquad (6)$$

6. Faktor δj ini digunakan untuk menghitung koreksi error (Δvij) yang nantinya akan dipakai untuk memperbaharui vij, di mana:

$$\Delta v_{ij} = \alpha \delta_j x_i \ldots\ldots\ldots\ldots \qquad (7)$$

7. Selain itu juga dihitung koreksi bias Δv0j yang nantinya akan dipakai untuk memperbaharui v0j , di mana:

$$\Delta v_{0j} = \alpha \delta_j \ldots\ldots\ldots\ldots \qquad (8)$$

8. Setiap unit output (Yk ,k=1,…,m) akan memperbaharui bias dan bobotnya dengan setiap hidden unit.

$$w_{jk}(baru) = w_{jk}(lama) + \Delta w_{jk} \ldots\ldots\ldots (9)$$

3. Hasil Percobaan

Percobaan yang menggunakan arsitektur Residual Network dilakukan beberapa kali dengan mengubah nilai learning rate, batch size, layer, dan jumlah epoch dalam proses pelatihan. Variasi dan hasil percobaan dari arsitektur ini adalah sebagai berikut:

Tabel 1. Variasi Percobaan dengan Arsitektur ResNet

No	Epoch	LR	Layer	Batch Size	val_acc
1	80	0.05	10	32	3.01
2	80	0.05	10	64	2.98
3	80	0.03	10	32	3.1
4	80	0.03	10	64	3.1
5	80	0.03	15	32	3.202
6	100	0.05	10	32	2.3
7	100	0.05	10	64	2.3
8	100	0.07	10	32	2.1
9	100	0.07	10	64	1.62
10	120	0.05	10	32	4.4
11	120	0.05	10	64	4.1
12	120	0.03	10	32	**12.61**
13	120	0.03	10	64	6.2
14	120	0.07	10	32	4.9
15	120	0.07	10	64	4.9
16	150	0.05	10	32	6.6
17	150	0.07	10	32	6.59

Seperti yang dapat dilihat pada Tabel 1, hasil percobaan menggunakan arsitektur Residual Network sangatlah tidak bagus. Hasil akurasi validasi yang paling besar yang didapatkan dari percobaan-percobaan tersebut hanyalah 12.1% yaitu milik percobaan 12.

Untuk menganalisis hasil dari *clasiffier CNN* dengan menggunakan arsitektur Residual Network, digunakan grafik loss serta tabel *classifier report* yang berisikan nilai *precision, recall,* dan *f1-score.*

Tabel 2. Classifier Report percobaan 12 ResNet

	precision	recall	f1-score
0	0.00	0.00	0.00
1	0.00	0.00	0.00
2	0.00	0.00	0.00
3	0.00	0.00	0.00
4	0.00	0.00	0.00
5	0.00	0.00	0.00
6	0.25	0.20	0.22
7	0.33	0.20	0.25
8	0.25	0.20	0.22
9	0.00	0.00	0.00
10	0.00	0.00	0.00
11	0.00	0.00	0.00
12	0.00	0.00	0.00
13	0.00	0.00	0.00
14	0.00	0.00	0.00
15	0.00	0.00	0.00
avg / total	0.05	0.04	0.04

Dari tabel diatas dapat diketahui bahwa hampir seluruh kelas memiliki nilai precision sebesar 0. Hanya terdapat 3 kelas yang memiliki nilai precision diatas 0, yaitu kelas Candi Ratu Boko, Candi Sewu, dan Museum Fatahillah. Hal ini dapat disebabkan oleh karena Candi Ratu Boko memang memiliki bentuk yang sangat berbeda dari kelas-kelas lain. Sedangkan candi Sewu memiliki kemiripan bentuk bangunan dengan candi Prambanan.

Gambar 1. Perbandingan Candi Prambanan (Kanan) dan candi Sewu (Kiri)

Meskipun demikian Candi Sewu masih memiliki nilai *precision,* Candi Prambanan tidak memiliki nilai *precision* dan tidak memiliki satu pun hasil prediksi yang benar. Hal ini dapat menandakan bahwa candi Sewu memiliki data pelatihan yang lebih baik dari pada candi Prambanan. Namun jika diperhatikan dari tabel classifier report percobaan lain, candi Prambanan memiliki nilai precision sebesar 0.25 sedangkan candi Sewu memiliki nilai precision 0. Hal ini membuktikan bahwa data pelatihan candi Sewu maupun candi Prambanan tidak buruk.

Untuk hasil yang menggunakan arsitertur Inception dengan variable bebas nilai learning rate, batch size, layer, jumlah data training, dan jumlah epoch dapat dilihat pada tabel berikut:

Tabel 3. Variasi Percobaan dengan Arsitektur Inception

No	Epoch	Learning Rate	Batch Size	Jumlah Data Training	Validation Accuracy
1	40	0.05	32	1024	19.1
2	40	0.05	64	1024	3.6
3	40	0.03	32	1024	19.9
4	40	0.03	64	1024	19.6
5	40	0.08	32	1024	25.1
6	50	0.05	32	1024	3.9
7	50	0.05	64	1024	25.5
8	50	0.07	32	1024	17.1
9	50	0.07	64	1024	17
10	70	0.05	32	1024	20.99
11	70	0.05	64	1024	20.98
12	70	0.03	32	1024	32.1
13	70	0.03	64	1024	34.3
14	70	0.07	32	1024	35.3
15	70	0.07	64	1024	33.2
16	100	0.05	16	1024	39.9
17	100	0.05	32	1024	**41.2**
18	100	0.05	64	1024	38.2
19	100	0.07	32	1024	40.1
20	100	0.05	32	1610	20.11

Sama halnya dengan untuk percobaan dengan ResNet, dilihat pula classification report yang dihasilkan oleh percobaan terbaik oleh inception (percobaan 17).

Pada percobaan 17 didapatkan hasil yang paling bagus diantara pecobaan-percobaan lainnya yaitu dengan hasil 41.1%. Pada percobaan ini pula disadari sebuah keganjilan yaitu jika pengujian untuk museum dan candi dipisahkan maka didapatkan hasil akurasi sebesar 72.9% untuk museum dan 25.2% untuk candi.

Setelah dilakukan analisis menggunakan confusion matrix ditemukan bahwa hasil prediksi untuk museum memang jauh lebih tepat dibandingkan dengan hasil prediksi yang didapatkan untuk candi. Kelas-kelas candi banyak mendapatkan hasil false negative dan false positive.

Untuk mengatasi hasil yang kurang memuaskan pada kelompok kelas candi, dilakukan percobaan 20, yaitu menambahkan jumlah data training untuk kelas-kelas candi. Pada percobaan ini, kelas-kelas museum masing-masing menggunakan 80 gambar, sedangkan untuk kelas candi masing-masing kelas menggunakan 140 gambar. Learning rate, jumlah layer, jumlah epoch, dan jumlah batch yang digunakan dalam percobaan ini menggunakan nilai yang sama dengan percobaan 17 karena percobaan 17 menghasilkan nilai yang paling maksimal diantara pecobaan-percobaan yang telah dilakukan sebelumnya.

Percobaan 20 menghasilkan hasil yang overfitting (suatu keadaan dimana model terlalu terpacu dengan data pelatihan yang digunakan untuk pelatihan), sehingga accuracy pada saat pelatihan dapat mencapat 87% namun accuracy yang didapatkan dari data validasi sengatlah kecil yaitu 19%. Keadaan ini membuat aplikasi ObSata tidak dapat memprediksi data yang benar-benar baru. Hasil yang overfitting ini dapat dikarenakan oleh beberapa hal yaitu epoch yang terlalu banyak untuk set data tersebut atau karena variasi gambar data pelatihan kurang baik.

Tabel 4. Classifier Report percobaan 17 Inception

	precision	recall	f1-score
0	0.30	0.43	0.35
1	0.15	0.29	0.20
2	0.67	0.29	0.40
3	0.40	0.29	0.33
4	0.20	0.14	0.17
5	0.40	0.29	0.33
6	0.40	0.29	0.33
7	0.60	0.43	0.50
8	0.38	0.43	0.40
9	0.22	0.29	0.25
10	0.38	0.86	0.52
11	0.29	0.29	0.29
12	0.80	0.57	0.67
13	0.80	0.57	0.67
14	0.67	0.57	0.62
15	0.80	0.57	0.67
avg / total	0.47	0.41	0.42

Pada tabel diatas dapat dilihat bahwa seluruh kelas memiliki nilai precision yang menandakan bahwa seluruh kelas mendapatkan hasil prediksi yang benar. Secara keseluruhan dapat dilihat pula bahwa kelas-kelas museum memiliki nilai precision yang relatif lebih baik dibandingkan kelas-kelas candi.

Hasil akhir yang digunakan untuk aplikasi final adalah percobaan 17 Inception. Percobaan tersebut dipilih sebagai hasil akhir karena memiliki precission yang cukup merata dan memiliki nilai akurasi yang lebih baik dari percobaan-percobaan lain.

4. Kesimpulan

Berdasarkan hasil pengujian yang telah dilakukan, maka kesimpulan yang didapat dari pembuatan Aplikasi Pencarian Objek Wisata Bersejarah di Pulau Jawa Menggunakan Convolutional Neural Network adalah sebagai berikut:

1. Website ObSata dapat menghasilkan nama, lokasi, dan sekilas informasi tentang objek wisata dengan input sebuah gambar tampak depan dari objek wisata.
2. Website ObSata dapat menghasilkan objek wisata sejenis dengan sesuai dengan pilihan lokasi.
3. Arsitektur Inception lebih cocok digunakan untuk kasus pengenalan bangunan.

Saran yang dapat membantu dalam mengembangkan dan meningkatkan kualitas sistem yaitu:

1. Menambah banyaknya objek wisata yang dapat ditemukan oleh sistem.
2. Dilakukan training dengan arsitektur lain atau variasi lain agar dapat menghasilkan akurasi yang lebih baik.

REFERENSI

[1] Christian S.; Vincent V.; Sergey I.; Shlens, Jon dan Sbigniew W. “Rethinking the Inception Architecture for Computer Vision”. International Conference on Image, Vision and Computing (ICIVC). Juni, 2012.

[2] Ciresan, Dan C.; Meier, Ueli.; Masci, Jonathan.; Gambardella, Luca Maria dan Schmudhuber, Jurgen. Flexible, High Performance Convolutional Networks for Image Classification. Barcelona: UJCAI, Juli, 2011.

[3] Haykin, Simon. Neural Network and Learning Machines. London: Pearson, 2009.

[4] He, Kaiming.; Zhang, Xiangyu.; Ren, Shaoqing dan Sun, Jian. “Deep Residual Learning for Image Recognition”. IEEE Conference on Computer Vision and Pattern Recognition. Salt Lake City: IEEE, Juni, 2016.

[5] Homik, Kurt.; Stichcombe, Maxwell dan White, Halbert. “Multilayer Feedward Network are Universal Approximators”. Jurnal Neural Networks, Vol 2, No. 5. Oxford: Pergamon Press, Oktober, 1989.

[6] Rosebrock, Adrian. ImageNet: VGGNet, ResNet, Inception, and Xception with Keras. https://www.pyimagesearch.com/2017/03/20/imagenet-vggnet-resnet-inception-xception-keras/. 4 September 2018.

[7] UFLDL Standford. Deep Networks: Overview. http://ufldl.stanford.edu/wiki/index.php/Deep_Networks:_Overview. 7 September 2018.

Nadia Ramadhani, Program Studi Teknik Informatika Fakultas Teknologi Inforsmasi Universitas Tarumanagara Tahun 2019

Janson Hendryli, Dosen Program Studi Teknik Informatika Fakultas Teknologi Inforsmasi Universitas Tarumanagara. Memperoleh gelar S. Kom dari Universitas Tarumanagara dan memperoleh gelar M. Kom dari Universitas Indonesia.

Dyah Erny Herwindianti, Dosen Program Studi Teknik Informatika Fakultas Teknologi Inforsmasi Universitas Tarumanagara. Memperoleh gelar sarjana dalam program studi Statistik, Matematika, dan Ilmu Pengerahuan Alam dari ITS Surabaya. Memperoleh gelar Magister Sains dari Institut Pertanian Bogor. Memperoleh gelar doktor dari program studi MIPA, Matematika, dan Ilmu Pengetahuan Alam dari Institut Teknologi Bandung.