

IMPLEMENTASI MACHINE LEARNING UNTUK MEMBUAT IMAGE SEARCH ENGINE BERDASARKAN ANALISIS SIMILARITAS

[IMPLEMENTATION OF MACHINE LEARNING TO CREATE AN IMAGE SEARCH ENGINE BASED ON SIMILARITY ANALYSIS]

Austin Darian Pratama¹, Junita^{2*}

^{1,2}Departemen Teknik Elektro, Universitas Pelita Harapan, Tangerang, Indonesia

*Korespondensi penulis: junita.fti@uph.edu

ABSTRACT

The development of technology makes it easier to search for something on the internet. Based on data in 2022, there will be more than 1 billion Google users worldwide. This data proves the importance of search engines for internet users. However, in general, search engine searches use keywords. This research aimed to produce image search engines for searching. The image search engine utilizes a Convolutional Neural Network, namely VGG-16, which would compare the similarity of the input image to all images in the database by comparing the neurons in the image. Image search engine performance was measured based on the accuracy and sequence of image output. Accuracy was obtained with a weighting system from 5 search result images produced by the image search engine. Mobile phone applications were used to improve the quality of images used for searching. The application would capture images or take images through the gallery then upload them to Firebase and be used to search by image search engines. From the research conducted, it was found that changing the angle had the greatest impact on decreasing accuracy values when compared to 3 other factors: color, background and image quality. The decrease in accuracy due to the influence of angle, color, background, and image quality was 96.25%, 0%, 22.5%, and 68.75%, respectively. The images from cellphone application were proven to have higher accuracy (95.32%) than the images captured with webcam (86.67%). The increase in accuracy emphasizes that the influence of image quality and the angle of the image used to search influences search results. The specifications of the cellphone camera are 64 Megapixels, while the webcam camera is 2 Megapixels.

Keywords: *convolutional neural network; firebase; image search engine; image similarity; mobile application*

ABSTRAK

Berkembangnya teknologi memberikan kemudahan untuk mencari sesuatu dalam internet. Berdasarkan data pada tahun 2022 lebih dari 1 miliar pengguna google di seluruh dunia. Data tersebut membuktikan bahwa pentingnya *search engine* bagi pengguna internet. Namun pada umumnya, pencarian *search engine* memanfaatkan kata kunci. Penelitian ini bertujuan untuk menghasilkan *image search engine* untuk pencarian. *Image search engine* memanfaatkan *Convolutional Neural Network* yaitu VGG-16 yang membandingkan kemiripan gambar *input* dengan semua gambar pada *database* dengan membandingkan neuron–neuron pada gambar. Performa *image search engine* diukur berdasarkan akurasi dan urutan keluaran gambar. Akurasi didapatkan dengan sistem pembobotan dari 5 gambar hasil pencarian yang dihasilkan oleh *image search engine*. Aplikasi handphone dimanfaatkan untuk meningkatkan kualitas gambar yang digunakan untuk pencarian. Aplikasi akan menangkap gambar atau mengambil gambar melalui *gallery* kemudian diunggah ke *firebase* dan digunakan untuk mencari oleh *image search engine*. Dari penelitian yang dilakukan didapatkan bahwa perubahan *angle* memberikan dampak yang paling besar terhadap

penurunan nilai akurasi jika dibandingkan dengan 3 faktor lainnya: warna, background, dan kualitas gambar. Penurunan akurasi yang didapatkan akibat pengaruh *angle*, warna, *background*, dan kualitas gambar masing-masing sebesar 96,25%; 0%, 22,5%; dan 68,75%. Gambar dari aplikasi *handphone* terbukti memiliki akurasi lebih tinggi (95.32%) dari gambar hasil tangkapan webcam (86,67%). Peningkatan akurasi ini mempertegas bahwa pengaruh kualitas gambar dan *angle* pada gambar yang digunakan untuk mencari mempengaruhi hasil pencarian. Spesifikasi dari kamera *handphone* adalah 64 Megapixels, sedangkan kamera webcam 2 Megapixels.

Kata kunci: *cConvolutional neural network*; *firebase*; mesin pencari gambar; kesamaan gambar; aplikasi seluler

PENDAHULUAN

Berkembangnya teknologi mendatangkan kemudahan bagi manusia. Salah satu manfaat yang dapat dirasakan adalah kemudahan untuk mendapatkan informasi yang relevan dengan menggunakan *search engine*. *Search engine* dapat berguna apabila informasi sesuai yang dihasilkan dengan preferensi pengguna. Berdasarkan data yang didapatkan, pada tahun 2022 lebih dari 1 miliar pengguna google membuktikan pentingnya *search engine* bagi pengguna internet. Data dari Pew Internet memperlihatkan bahwa 91% dari pengguna program pencari Google menyatakan bahwa mereka selalu mendapatkan hal yang mereka cari dan 73% dari pengguna menyatakan bahwa informasi yang mereka dapatkan dapat dipercaya (Purcell *et al.*, 2012). *Image search engine* merupakan salah satu jenis *search engine* yang digunakan untuk mencari informasi mengenai suatu objek dengan cara

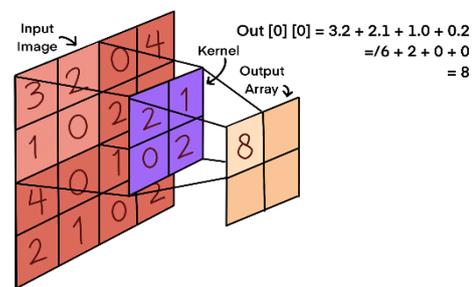
menangkap gambar dari objek yang ingin dicari. Salah satu keuntungan yang didapatkan adalah *image search engine* memberi kemudahan untuk tidak perlu mengetahui serta mengenal objek yang ingin dicari. Dari objek yang dilihat, pengguna dapat langsung melakukan pengambilan gambar untuk mengetahui serta mengenal objek yang dicari. Jadi peran penting *image search engine* adalah untuk menghasilkan informasi yang relevan bagi pengguna sesuai dengan gambar yang digunakan untuk mencari (Sini & Ata, 2013).

Pada penelitian ini dibuat *search engine* yang memanfaatkan gambar untuk mencari atau *image search engine*. *Image search engine* ini dirancang untuk dapat memberikan atau menampilkan gambar yang serupa dengan objek yang dicari. Dengan membandingkan neuron-neuron gambar, *image search engine* yang dirancang mencari kemiripan antara gambar dari objek yang dicari dengan semua gambar yang tersimpan pada

database. *Image search engine* memanfaatkan VGG-16 yang merupakan *convolutional neural network* yang sudah melewati proses *training*. Disebut VGG-16 dikarenakan memiliki 16 *layers* yang mengandung *weight* (Ganesh, 2020). *Weight* adalah sebuah nilai yang dikaitkan dengan *features* di mana nilai tersebut mewakili atau menunjukkan seberapa penting *features* tersebut (Dongare *et al.*, 2012). *Features* bisa juga dikatakan sebagai informasi, untuk kasus ini informasi pada gambar. Untuk mendapatkan *convolutional layer* diperlukan *Kernel Filter*. *Kernel Filter* merupakan sebuah filter yang berfungsi untuk melihat informasi secara lokal atau pada area tertentu. Filter ini mengandung nilai *weight* yang disusun dalam bentuk matriks. Hasilnya akan membentuk *convolutional layer* dengan dimensi informasi yang lebih kecil namun mewakili informasi dari input atau disebut ekstrak (Hijazi *et al.*, 2015). Namun perlu diingat bahwa *convolutional layer* tidak mengurangi dimensi dari layer, maka perlu dilakukan *padding* untuk menjaga agar ukuran layer tetap sama. Contoh dari proses *feature extraction* dapat dilihat pada Gambar 1.

Tujuan dari *feature extraction* adalah mengurangi beban dari mesin untuk melakukan komputasi tanpa mengurangi

kualitas informasi yang dibawa. Cara lainnya adalah menggunakan *pooling layer*, yang juga digunakan pada VGG-16. Terdapat 2 jenis *pooling layer*, yaitu *maximum pooling layer* dan *average pooling layer* (Zafar *et al.*, 2022). Pada VGG-16 digunakan *maximum pooling layer*, dimana akan mengambil nilai informasi terpenting pada suatu lokal.



Gambar 1. Contoh dari proses *feature extraction*

Program ini dibuat dengan bahasa pemrograman Python karena Python merupakan *high level programming language* yang menitikberatkan pada keterbacaan kode sehingga mudah dipahami (Srinath, 2017). Selain itu, *packages* yang tersedia pada Python juga menunjang *image search engine* yang hendak dibuat seperti seperti Numpy, Scikit-learn, Matplotlib, Sci-py, dan OpenCV. Python juga menyediakan fitur PIP atau *Python Package Manager* dimana *programmer* mampu melakukan instalasi *package* yang diinginkan (Bogdanchikov *et al.*, 2013). Python juga disebut *portable code* yang artinya kodenya dapat dijalankan di

operating systems atau mesin yang berbeda seperti contohnya Mac atau Linux (Gowrishankar & Veena, 2018).

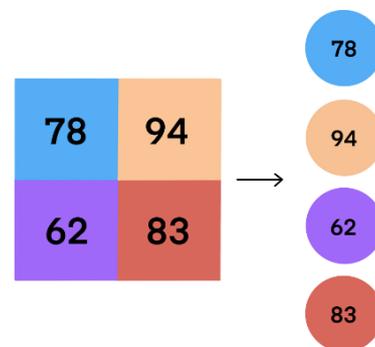
Pada penelitian ini juga dilakukan analisis performa dari program. Analisis performa dilakukan dengan melihat pengaruh gambar *input* terhadap hasil pencarian. Gambar *input* yang digunakan bervariasi dari *angle*, kualitas gambar, warna, dan *background*. Selain itu, pada penelitian ini, penulis juga membuat aplikasi telepon genggam yang digunakan untuk mengambil gambar serta mengunggah gambar yang ditangkap menuju *firebase*. *Firebase* mendukung integrasi fitur *cloud-based* pada aplikasi *mobile* yang memungkinkan *user* untuk menyimpan data secara *real-time* (Smyth, 2017). Gambar yang ditangkap oleh aplikasi digunakan sebagai gambar referensi untuk pencarian.

BAHAN DAN METODE

Metode Penelitian

Pertama, dibentuk program pencari gambar. Penelitian ini memanfaatkan VGG-16 yang merupakan CNN dari library Keras. Kemudian, digunakan *google colab* untuk menjalankan program pencari gambar yang memiliki bahasa pemrograman Python versi 3.8. Untuk melihat kemiripan antara gambar akan dilakukan perbandingan hasil *feature*

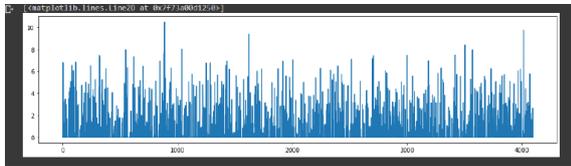
extraction dari layer *fc2*. Layer *fc2* merupakan lapisan *fully connected layer*. *Fully connected layer* pada CNN digunakan untuk proses klasifikasi. Hal ini dikarenakan operasi perkalian matriks yang akhirnya akan menghasilkan sebuah *output* dari *fully connected layer* yang merepresentasikan semua *input*-nya. Contohn dapat dilihat pada Gambar 1.



Gambar 2. *Flattened input vector*

Dari Gambar 2, *input vector* yang diterima dari *convolutional layer* menuju *fully connected layer* berbentuk matriks 2 x 2. Matriks tersebut diratakan menjadi matriks 1 x 4. Matriks 1 x 4 tersebut dikalikan dengan matriks *weight* yang nilainya didapatkan melalui hasil *training*. VGG-16 merupakan *pre-trained CNN* sehingga *weight* pada matriks beban sudah terdapat nilainya tanpa diperlukan *training* ulang pada penelitian ini. Bentuk dari matriks beban apabila memanfaatkan input *vector* dari Gambar 2 adalah 4 x 4096, sehingga hasil dari *fc2* adalah matriks 1 x 4096. *Output* dari *fc2* tersebut digunakan untuk membandingkan gambar *input*

dengan gambar pada *database*. Hasil *feature extraction* yang telah di-plot dapat dilihat pada Gambar 3.



Gambar 3. Hasil *feature extraction*

Hasil ini disederhanakan dengan *principal component analysis*, sehingga 4096 *features* mampu disederhanakan menjadi 300 *features*. Mengoperasikan lebih dari 4096 elemen tidak efisien dalam hal kebutuhan ruang atau memori dan kecepatan prosesor, oleh karena itu digunakan PCA untuk mengurangi panjang vektor namun tetap mempertahankan representasi efektif yang sama.

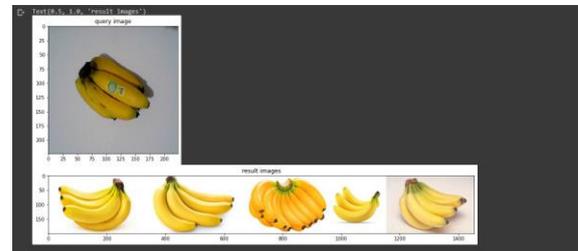
Selanjutnya adalah menyediakan *database* gambar yang menjadi tempat penampungan gambar. *Database* terdiri dari 445 gambar (Gambar 4).



Gambar 4. *Database* gambar

Gambar dari objek yang hendak dicari akan diproses oleh program pencari gambar sehingga dihasilkan hasil *feature extraction*-nya. Lalu, hasil *feature extraction* dari gambar objek dibandingkan dengan semua hasil *feature extraction* pada

masing-masing gambar. Gambar dengan kemiripan yang tinggi ditampilkan sebagai hasil (Gambar 5).



Gambar 5. Contoh hasil pencarian

Gambar objek yang hendak dicari yaitu pisang ditampilkan sebagai *query image* dan 5 gambar pisang lainnya ditampilkan sebagai hasil pencarian. Gambar paling kiri dari lima gambar yang ditampilkan sebagai *result images* memiliki tingkat kemiripan paling tinggi diantara lima gambar yang ditampilkan. Oleh karena itu, untuk perhitungan akurasi dapat dilihat pada Gambar 6.



Gambar 6. Simulasi perhitungan akurasi

Skor yang didapatkan dibagi 5 untuk mendapatkan nilai akurasi pada

setiap pencarian. Pada penelitian kali ini dilakukan analisis performa dari *image search engine*. Analisis memperhatikan beberapa faktor. Pertama, *angle* objek pada gambar. Tujuannya adalah untuk memperhatikan pengaruh *angle* terhadap hasil pencarian. Kedua, adalah warna dari objek pada gambar yang bertujuan untuk memperhatikan pengaruh warna objek yang digunakan sebagai bahan untuk pencarian pada hasil pencarian. Ketiga, pengaruh dari background atau keberadaan objek lain dari objek yang digunakan sebagai bahan untuk pencarian pada hasil pencarian. Keempat, adalah percobaan melihat pengaruh kualitas gambar yang digunakan untuk mencari pada akurasi. Kualitas gambar dibuat bervariasi dengan cara kompresi pada gambar yang digunakan untuk mencari. Kelima, diperlihatkan perbedaan memanfaatkan gambar *input* dari aplikasi *handphone* dengan gambar *input* hasil tangkapan *webcam*. Bagian kelima berkaitan dengan bagian keempat yaitu kualitas gambar *input* yang digunakan untuk mencari.

HASIL DAN PEMBAHASAN

Pertama, untuk faktor *angle* pada objek yang dicari. *Angle* yang buruk dapat mengakibatkan objek yang tertangkap kamera terlihat seperti objek yang berbeda dari *database*. Dicoba kepada 4 objek

dengan 4 *angle* yang berbeda. *Angle* pertama dari *angle* normal, kedua dari *angle* kanan atau kiri, ketiga serong dari objek, dan terakhir dari atas objek. Tabel 1 memperlihatkan bahwa akurasi turun sebesar 96,25% dari akurasi maksimal yang dapat dicapai pada *angle* normal. *Angle* kanan atau kiri dari objek memberikan dampak paling besar pada penurunan akurasi akibat *angle*.

Tabel 1. Pengaruh *angle* pada akurasi hasil pencarian

Objek	Akurasi				Selisih
	1	2	3	4	
Objek 1	100	25	40	15	85
Objek 2	100	0	85	0	100
Objek 3	100	0	0	35	100
Objek 4	90	15	0	0	100
			Rata - rata		96,25

Kedua, adalah faktor warna. Pada *database* gambar tersedia gambar objek yang sama namun memiliki variasi warna. Percobaan kali ini ingin melihat apakah terdapat pengaruh perbedaan warna pada objek yang dicari pada hasil pencarian.

Tabel 2. Pengaruh perbedaan warna pada objek pada hasil pencarian

Objek	Akurasi			Selisih
	1	2	3	
Objek 1	100	100	100	0
Objek 2	100	100	100	0
Objek 3	100	100	100	0
Objek 4	100	100	100	0
			Rata - rata	0

Tabel 2 menunjukkan bahwa perbedaan warna dari objek yang digunakan untuk mencari dengan warna dari objek pada *database* tidak mempengaruhi hasil pencarian.

Ketiga, adalah faktor *background*. *Background* gambar pada objek mempengaruhi hasil pencarian. *Background* diibaratkan sebagai *noise* atau gangguan pada hasil pencarian. *Background* yang penuh dengan objek lain akan menurunkan akurasi dari hasil pencarian. Pengaruh *background* yang memiliki objek lain di luar objek yang sedang dicari mampu menurunkan akurasi dari program pencari sebesar 22,5% (Tabel 3).

Tabel 3. Pengaruh *background* pada hasil pencarian

Objek	Akurasi			Selisih
	1	2	3	
Objek 1	100	90	100	10
Objek 2	100	100	100	0
Objek 3	100	60	70	40
Objek 4	90	50	70	40
	Rata - rata			22.5

Keempat, adalah faktor kualitas gambar *input*. Penelitian dilakukan dengan memakai gambar yang sama sebagai gambar *input* namun dilakukan degradasi kualitas gambar yang digunakan untuk pencarian. Penurunan kualitas gambar dilakukan dengan metode kompresi yang diterapkan pada gambar. Semakin buruk kualitas gambar atau semakin tinggi kompresi yang dilakukan maka semakin menurun akurasinya, hingga mencapai 68,75% (Tabel 4). Semakin besar kompresi yang dilakukan semakin buruk akurasi dari program pencari gambar. Artinya semakin buruk kualitas gambar yang digunakan

untuk mencari, maka hasil pencarian semakin tidak akurat.

Dari keempat faktor yang diuji, Faktor yang mempengaruhi penurunan akurasi terbesar adalah faktor *angle* dan kualitas gambar. Maka dari itu, dimanfaatkan aplikasi telepon genggam yang dikembangkan sendiri dengan fitur menangkap gambar dan mengunggah gambar ke *firebase*. Keuntungan yang didapatkan adalah kualitas gambar yang dihasilkan jauh lebih baik dibandingkan dengan gambar tangkapan *webcam* dan *angle* gambar dari objek dapat disesuaikan. Hasil akurasi program pencari gambar setelah memanfaatkan hasil tangkapan *mobile application* sebagai *input* meningkat jika dibandingkan dengan hasil akurasi program pencari gambar ketika memanfaatkan hasil tangkapan dari *webcam*. Akurasi yang dihasilkan meningkat dari 86,67% menjadi 95,33% (Tabel 5). Hal ini dikarenakan kualitas gambar dan *angle* gambar objek dari *mobile application* lebih baik dibandingkan gambar dari *webcam*.

Tabel 4. Pengaruh kualitas gambar yang dikompresi pada hasil pencarian

	Akurasi			Selisih
	0%	50%	90%	
Objek 1	100	35	25	75
Objek 2	100	35	20	80
Objek 3	100	70	45	55
Objek 4	100	55	35	65
	Rata - rata			68,75

Tabel 5. Perbandingan hasil akurasi *mobile application* dan *webcam*

<i>Mobile Application</i>				<i>Webcam</i>							
No.	Objek	Akurasi (%)	Waktu Proses (detik)	No.	Objek	Akurasi (%)	Waktu Proses (detik)	No	Objek	Akurasi (%)	Waktu Proses (detik)
1.	Monitor	100	0.890	24.	Sabun	100	2.647	1.	Handphone	80	0.89
2.	Gagang pintu	100	1.953	25.	Dispenser selotip	100	1.223	2.	Hard disk	85	2.561
3.	Baju	100	2.203	26.	Sapu	100	0.976	3.	Pulpen	100	1.51
4.	Botol	100	1.742	27.	Tanaman	100	2.931	4.	Mouse	100	1.774
5.	Cukuran	100	1.283	28.	Tas	100	2.390	5.	Baju	70	2.05
6.	Dispenser	70	1.387	29.	Tempat sampah	70	1.328	6.	Sikat gigi	25	3.052
7.	Sabun cuci muka	100	1.379	30.	Toilet	60	1.707	7.	Handphone	75	1.274
8.	Galon	100	1.710	31.	Wastafel	100	0.936	8.	Tempat sampah	100	1.95
9.	Guling	55	2.005	32.	Tote bag	100	0.999	9.	Pisang	100	1.825
10.	Handphone	85	1.557	33.	Baygon	100	1.274	10.	Tanaman	100	2.21
11.	Hard disk	90	0.900	34.	Celana	100	1.617	11.	Kipas angin	100	1.3
12.	Keyboard	100	1.462	35.	Kuas	100	1.990	12.	Sapu	100	1.44
13.	Kipas angin	100	1.733	36.	Barbel	100	1.017	13.	Kursi kantor	75	1.36
14.	Kunci	100	1.377	37.	Kabel roll	100	1.579	14.	Sabun cuci muka	100	2.52
15.	Kursi	100	1.421	38.	Jeruk	100	2.064	15.	Speaker	100	2.33
16.	Pisang	100	1.529	39.	Sendal	100	1.834	16.	Keyboard	100	1.66
17.	Speaker	100	1.209	40.	Sapu lidi	55	1.721	17.	Remote tv	65	1.94
18.	Sofa tidur	100	1.352	41.	Teflon	100	1.972	18.	Galon	100	1.35
19.	Monitor	100	0.983	42.	Piring	100	2.040	19.	Cukuran	80	1.29
20.	Mouse	100	1.833	43.	Kursi roda	100	2.162	20.	Tumbler	65	1.843
21.	Printer	100	3.215	44.	Remote tv	100	1.201	21.	Sendal	100	0.92
22.	Saklar	100	2.089	45.	Sofa	100	1.485				
23.	Pulpen	100	1.036	46.	Sepatu	100	1.340				
Average						95.326	1.624	Average		86.667	1.764

KESIMPULAN

VGG-16 mampu dimanfaatkan sebagai CNN untuk *image search engine* dan mampu menghasilkan akurasi 95,33%. Penurunan akurasi terbesar dari *image*

search engine dipengaruhi oleh *angle* sebesar 96,25% dan kualitas gambar sebesar 68,75%. Dengan aplikasi telepon genggam, program pencari gambar dapat mencapai akurasi yang lebih tinggi.

DAFTAR PUSTAKA

- Bogdanchikov, A., Zhaparov, M., & Suliyev, R. (2013). Python to learn programming. *Journal of Physics: Conference Series*, 423(1), 012027. <https://doi.org/10.1088/1742-6596/423/1/012027>
- Dongare, A., Kharde, R. R., & Kachare A. D. (2012). Introduction to artificial neural networks. *International Journal of Engineering and Innovative Technology*, 2(1), 189-194.
- Ganesh, K. S. (2020). *What's the role of weights and bias in a neural network?* Retrieved July 25, 2022 from <https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f>
- Gowrishankar, S., & Veena, A. (2018). *Introduction to python programming*. CRC Press. <https://doi.org/10.1201/9781351013239>
- Hijazi, S., Kumar, R., & Rowen, C. (2015). *Using convolutional neural networks for image recognition*. Retrieved October 20, 2022 from https://site.eet-china.com/webinar/pdf/Cadence_0425_webinar_WP.pdf
- Purcell, K., Brenner, J., & Rainie, L. (2012). *Search engine use 2012*. Retrieved May 25, 2022, from https://www.ris.org/uploadi/editor/1341041853PIP_Search_Engine_Use_2012.pdf
- Sini, M. A., & Abuata, B. A. (2013). Web image search engine evaluation. *International Arab Journal of Information Technology*, 3(2), 90-98.
- Smyth, N. (2017). *Firestore essentials - Android edition*. Payload Media.
- Srinath, K. R. (2017). Python—the fastest growing programming language. *International Research Journal of Engineering and Technology*, 4(12), 354-357.
- Zafar, A., Aamir, M., Nawi, N. M., Arshad, A., Riaz, S., Alruban, A., Dutta, A. K., & Almotairi, S. (2022). A comparison of pooling methods for convolutional neural networks. *Applied Sciences*, 12(17), 8643. <https://doi.org/10.3390/app12178643>