

# Pengembangan Sistem Pencarian Gambar Produk E-Commerce dengan *Convolutional Neural Network*

Ahmad Faishol Huda

Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Pamekasan, Indonesia  
afaisholh@gmail.com

Rinaldi Munir

Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung  
Bandung, Indonesia  
rinaldi@informatika.org

**Abstract**—Teknologi sistem pencarian gambar atau *Visual search* adalah salah satu teknologi yang ikut berkembang seiring perkembangan teknologi di bidang *computer vision*. Pada prosesnya teknologi ini membandingkan fitur-fitur dalam sebuah gambar dan menghitung tingkat kedekatan antar gambar lalu mengembalikan gambar sesuai urutan tersebut. Pembangunan sistem ini bisa dilakukan dengan mendesain arsitektur sendiri maupun menggunakan metode *transfer learning*. Pada penelitian ini, dilakukan eksperimen terhadap empat buah *pretrained model* dengan metode *transfer learning*, yaitu VGG, Inception, Resnet, dan MobileNet, dan satu buah model dengan desain arsitektur sendiri. Eksperimen dilakukan untuk menentukan arsitektur terbaik dan modul yang bisa menampahi performa sistem pencarian gambar serta bisa mengetahui kelebihan dan kelemahan setiap model. Model terbaik dijadikan sebagai basis pembangunan sistem pencarian gambar. Hasil eksperimen menunjukkan model Resnet mendapatkan hasil yang paling bagus yaitu prediksi kategori, presisi, dan *recall* sebesar 0.8640, 0.8912, 0.5779. Beberapa modul yang bisa membantu performa sistem adalah modul augmentasi data dan juga modul *reindexing*.

**Kata kunci;** Sistem pencarian gambar; *architecture*; *transfer learning*; modul;

## I. PENDAHULUAN

Pada era digital seperti sekarang ini dimana banyak perusahaan berbasis teknologi terutama perusahaan di bidang e-commerce seperti Bukalapak berkembang pesat perlu adanya teknologi pencarian yang cepat dan canggih. Mesin pencarian ini yang awalnya hanya bisa dilakukan dengan sebuah masukan teks perlu dikembangkan lagi mengikuti trend dan kebutuhan kehidupan orang yang sudah mempunyai gawai untuk menangkap dan mengolah gambar di sekitarnya. Teknologi tersebut adalah mesin pencari berdasarkan gambar atau lebih dikenal dengan *Visual search*.

*Visual search* adalah teknologi yang menerima gambar sebagai masukan dibanding dengan menggunakan input text seperti search pada umumnya. Sebuah sistem *visual search* akan mengembalikan koleksi gambar yang mirip dengan gambar yang diberikan (Houdong Hu dkk, 2015). *Visual search* sebelum era *machine learning* sulit dilakukan karena informasi yang bisa didapat dari gambar sangat sedikit. Informasi tersebut hanya berupa edge (bentuk), warna, dan semacamnya. Informasi tersebut biasa kita sebut *local feature*,

artinya fitur yang bisa dilihat oleh manusia secara langsung. Namun setelah adanya *machine learning* khususnya *deep learning* kita tidak hanya bisa mendapatkan *local feature* dari sebuah gambar namun juga *deep feature* dari gambar tersebut. Fitur inilah yang membuat *visual search* menjadi dapat dilakukan dengan lebih cepat dan akurat.

Penggunaan *machine learning* maupun *deep learning* sudah menjadi hal yang wajar pada saat ini. Ketersediaan library untuk digunakan dalam melakukan pelatihan model untuk tujuan tertentu sangat mempermudah orang dalam melakukan *machine learning* maupun *deep learning*. Hal ini membuka banyak ide untuk bisa dikembangkan menggunakan cara tersebut. Salah satu ide yang sekarang banyak diteliti dan diterapkan adalah *visual search*.

Sudah dilakukan beberapa eksperimen/penelitian terhadap topik *visual search* ini. Dari hasil penelitian-penelitian sebelumnya baik dari Yanhao Zhang dkk (2018) dalam papernya *Visual search* at Alibaba telah diketahui topologi yang menghasilkan model dengan akurasi terbaik adalah topologi *Convolutional Neural Network* dengan pembelajaran *deep learning*. topologi ini memanfaatkan *deep learning* untuk mengekstraksi fitur-fitur dari gambar. Fitur-fitur yang diekstraksi bukan hanya fitur lokal tapi juga *deep* fitur yang nantinya akan sangat mempengaruhi keakuratan dari model. Arsitektur dari *CNN*-nya sendiri merupakan hal yang sangat penting dalam pembuatan sebuah model pencarian gambar. Dalam arsitektur itu sendiri bisa berisi beberapa modul yang akan diimplementasikan. Misal Yanhao Zhang dkk (2018) menggunakan beberapa modul seperti *category prediction*, *joint detection*, and *feature learning*, dan *indexing and retrieval* dalam arsitektur model yang mereka buat sendiri. Sedangkan Houdong Hu dkk (2015) menggunakan modul *Image Representation and Features* yang berisi *two-step object detection and localization* yang hasilnya sangat meningkatkan hasil pencarian. Oleh karena itu, dalam pembuatan model *visual search* ini akan dilakukan eksperimen terhadap beberapa arsitektur dan juga modul-modul didalamnya dan nantinya akan dibangun sebuah model dengan arsitektur dan modul yang terbaik. Topologi yang akan digunakan adalah *CNN* sedangkan beberapa modul yang pasti merupakan core modulnya adalah *feature extraction*, *indexing* dan *retrieval*.

## II. PENELITIAN TERKAIT

### A. Visual search at Alibaba

Pada penelitian ini topologi yang digunakan adalah Deep Convolutional Neural network (CNN). Dalam pengaplikasiannya dilakukan Deep Metrik Embedding untuk mengukur kedekatan dari dua buah gambar yang sedang dibandingkan. Weakly supervised object localization juga dilakukan untuk mengidentifikasi objek yang terdapat di gambar yang diberikan. Fitur ini nantinya akan digunakan untuk mendapatkan kategori dari gambar yang diberikan. Nantinya kategori ini akan mengurangi proses perbandingan gambar dengan data gambar yang sudah ada. Joint detection and Feature learning dilakukan untuk mendeteksi objek dan mendapatkan fitur-fitur dari objek itu. Setelah itu akan dilakukan index reranking berdasarkan kedekatan data gambar ke gambar yang diterima. Model yang dihasilkan dari penelitian ini mendapatkan nilai recall@1 sebesar 0.415. Hasil ini lebih bagus jika dibandingkan dengan versi original nya yaitu sebesar 0.067. Model ini juga memiliki recall yang lebih bagus daripada beberapa model lain, misalnya Resnet50 sebesar 0.108 dan ResNet101 sebesar 0.128.

Kelebihan model dalam penelitian ini adalah model bisa melakukan ekstraksi fitur sesuai dengan kebiasaan klik dari pengguna. Kelebihan lain adalah penggunaan binary indexing sehingga mengurangi deployment resources dan meningkatkan kualitas interaksi pengguna dengan sistem.

### B. Visual search at Pinterest

Pada penelitian ini topologi yang digunakan adalah Convolutional Neural network (CNN) menggunakan Caffe framework. Salah satu hal yang unik pada penelitian ini adalah dilakukannya lokalisasi terlebih dahulu sebelum melakukan pencarian gambar. Gambar akan ditentukan dulu kelas nya, misal dalam hal ini adalah kategorinya sehingga nantinya pencarian tidak perlu dilakukan terhadap semua gambar tapi hanya pada kategori yang dihasilkan tersebut. Hal ini akan mempercepat pencarian dan juga mungkin meningkatkan akurasi dari pencarian. Sistem pencarian gambarnya sendiri juga menggunakan sistem terdistribusi sehingga bisa melakukan beberapa proses secara paralel. Hasil dari penelitian ini adalah sebuah sistem pencarian gambar dimana pengguna bisa memilih objek dalam gambar untuk dijadikan sebagai masukan kedalam model.

## III. DATASET

Dataset yang digunakan adalah dataset yang disediakan oleh Bukalapak. Dataset ini terdiri dari 7556 gambar dan 8 kategori. Persebaran data di setiap kategori adalah sebagai berikut:

Tabel I. Tabel Persebaran Data di Setiap Kategori

Kategori	Jumlah data
Baju (Kaos)	964
Celana (Jeans)	984
Dompot	962
Jam tangan	993
Kaos tangan	692
Sandal/Sepatu	998

Kacamata	972
Topi	991

## IV. ANALISIS DAN RANCANGAN SOLUSI

### A. Analisis Masalah

*Visual search* dalam pengaplikasiannya mempunyai banyak masalah dan juga tantangan. Selain masalah-masalah umum yang sudah ditemukan pada penelitian-penelitian sebelumnya, juga terdapat masalah yang muncul saat pembangunannya sendiri. Pada penelitian-penelitian sebelumnya sebuah model *Visual search* terdapat beberapa masalah yang sudah ditemukan, yaitu maintenance heterogeneous data, Yanhao Zhang et al. (2018), dan masalah cost effective, Yushi Jing et al. (2015) dan Houdong Hu et al. (2018). Sedangkan pada pembangunannya sendiri masalah bisa muncul baik dari dataset yang digunakan saat pembangunan, jenis model yang dipakai, dan juga hyperparameter yang dipilih.

Masalah pertama dari penelitian-penelitian sebelumnya adalah maintenance heterogeneous data. Proses training memerlukan data yang cukup banyak, Untuk menyimpan data dan setiap meta data nya diperlukan proses yang sangat lama dan cukup mahal. Masalah selanjutnya adalah cost effective, selain bisa mengolah dataset cukup besar, model juga harus bisa melakukan komputasi yang efisien. Faktor yang mempengaruhi masalah sebelumnya adalah proses data preprocessing dan juga ekstraksi fitur. pemilihan metode preprocessing yang tepat akan membuat model bisa melakukan training lebih cepat dan menambah akurasi akhir dari model tersebut. Ekstraksi fitur sendiri merupakan hal paling penting dalam pelatihan model. Ekstraksi fitur yang baik dari sebuah gambar akan menambah akurasi akhir dari model.

Permasalahan yang bisa muncul saat melakukan eksperimen salah satunya adalah dair dataset, permasalahan yang bisa muncul adalah masalah overfitting dikarenakan dataset yang dan digunakan sekarang cukup kecil (~7k data). Faktor lain yang juga mempengaruhi pelatihan model adalah algoritma/arsitektur yang dipakai dalam pelatihan model tersebut. Untuk mendapatkan hasil terbaik arsitektur yang dipakai harus sesuai dengan tujuan dari model tersebut. Hyperparameter juga sangat mempengaruhi suksesnya pembuatan model, hyperparameter yang baik didapat dari eksperimen berulang sampai mendapatkan nilai hyperparameter yang terbaik.

### B. Analisis Metode Pembangunan Model

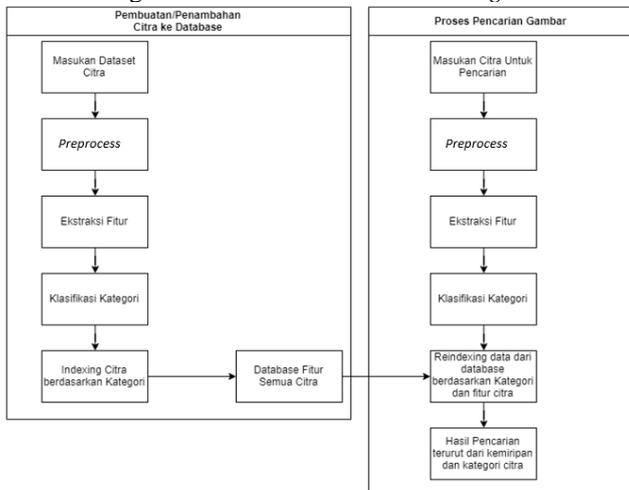
Salah satu algoritma atau topologi yang biasa digunakan dalam pembangunan sistem pencarian gambar adalah *Convolutional Neural Network*. Algoritma ini sudah sering dipakai oleh perusahaan-perusahaan besar seperti Alibaba, Pinterest, dan Ebay. Namun Melakukan pelatihan dengan menggunakan algoritma *CNN* memerlukan banyak data dan waktu yang lama. Untuk itu juga akan dicoba beberapa model yang sudah dilakukan training sebelumnya (pretrained) seperti beberapa model lain untuk melakukan *transfer learning*. Kita akan menggunakan beberapa model yang sudah ada

sebelumnya untuk melakukan *transfer learning*.. Beberapa model tersebut adalah GoogLeNet yang dibuat oleh Christian Szegedy dkk (2015), ResNet50 yang dibuat oleh Kaiming He dkk (2016), dan MobileNet. Kita akan mencoba beberapa dari arsitektur tersebut.

Selain akan mencoba menggunakan *transfer learning* akan dibuat juga model yang dibangun sendiri. Dari beberapa permasalahan sebelumnya diperlukan sebuah solusi untuk membuat sebuah model yang bisa menerima data yang *heterogeneous dalam volume* yang banyak dan juga tetap *cost-effective*. Pemilihan-pemilihan modul tersebut akan ditujukan untuk menyelesaikan beberapa permasalahan yang ada. Untuk permasalahan utama tentang *heterogeneous* data akan dilakukan supervised training terhadap model sehingga dataset menjadi tidak terlalu *heterogeneous*, lalu ada modul klasifikasi untuk nantinya klasifikasi data baru yang akan ditambah kedalam database sesuai dengan kategori yang sudah ada. Penggunaan modul *reindexing* akan mengurangi beban *resources* pada computer saat menggunakan model, metode *reindexing* yang akan dipakai adalah *KNearest Neighbour*. Metode ini nantinya akan membuat *index* untuk fitur dari setiap data dengan mengukur kedekatan masing masing data dengan metrik cosine dan nantinya saat proses pencarian sistem hanya perlu melakukan look up terhadap index ini dengan query yang ada. Dataset yang ukurannya terlalu kecil perlu dilakukan augmentasi data, sudah ada beberapa library yang menyediakan hal ini.

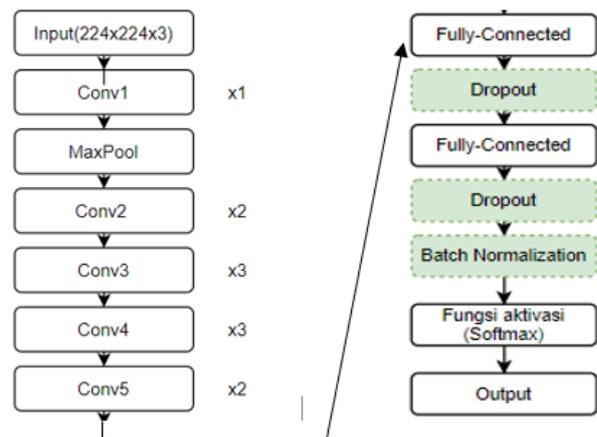
### C. Solusi Sistem Pencarian Gambar

Solusi arsitektur sistem yang akan digunakan adalah sebagai berikut. Sistem akan terbagi menjadi 2 bagian besar, yaitu bagian untuk penambahan gambar kedalam database dan bagian pencarian gambar. Pada bagian penambahan gambar, Dataset akan di *preproses*, setelah itu untuk setiap gambar dilakukan ekstraksi fitur dan klasifikasi. Fitur-fitur gambar akan disimpan dalam database. Pada bagian pencarian gambar, gambar akan di *preprocess* lalu melalui ekstraksi fitur, dan klasifikasi kategori. Hasil fitur akan dibandingkan dengan semua fitur gambar di database untuk *indexing*.



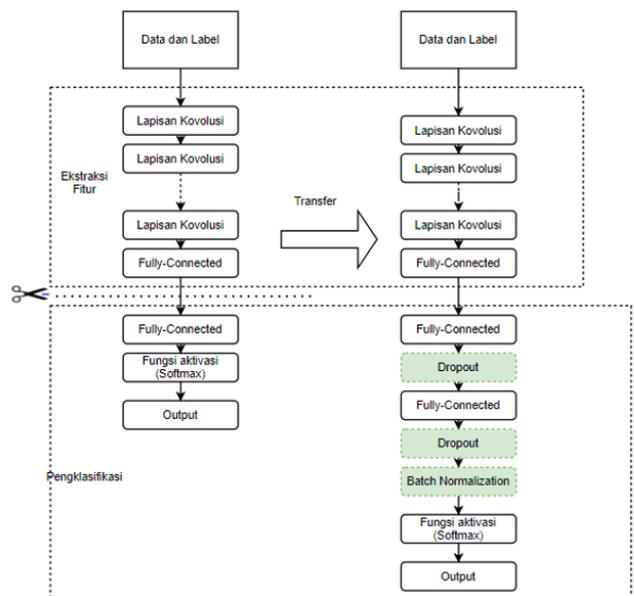
Gambar I Rancangan Arsitektur Sistem Pencarian Gambar

Arsitektur yang akan dipakai untuk model sendiri adalah sebagai berikut:



Gambar II Arsitektur Model Buatn Sendiri

Sedangkan untuk model yang memakai teknik transfer learning memiliki arsitektur yang berbeda dengan model buatan sendiri. Arsitektur yang digunakan akan menggabungkan pretrained model dengan pengklasifikasi buatan sendiri, gambarannya adalah sebagai berikut:



Gambar III Arsitektur Model dengan Metode *Transfer learning*

## V. EKSPERIMEN

### A. Kakas yang digunakan

Pada eksperimen kali ini kakas-kakas yang digunakan adalah, Google Colabs, Jupyter Notebook, Tensorflow, dan Keras. Bahasa pemrograman yang dipakai adalah Bahasa Python.

### B. Konfigurasi Hyperparameter

Pada eksperimen untuk pelatihan model sendiri dilakukan dengan menggunakan beberapa *hyperparameter* dalam pelatihan modelnya. Berikut adalah *hyperparameter* yang digunakan:

- *Epoch* : 20 - 300
- *Learning rate* ( $\alpha$ ) : 0.00001-0.001
- *Optimizer* : Adam
- *Fine Tuning Layer* : 5-30

Untuk setiap model akan dicoba dilatih dengan hyperparameter diatas, lalu akan diambil hasil terbaik dari setiap model untuk nantinya dibandingkan dan dianalisis performanya.

### C. Preprocess Dataset

Sebelum dataset digunakan untuk melakukan pelatihan model, dataset perlu dilakukan *preprocessing* terlebih dahulu, beberapa langkah *preprocessing* yang dilakukan adalah, melakukan pembersihan terhadap dataset agar semua gambar sesuai dengan kategorinya, setelah itu dilakukan pelabelan manual terhadap semua data. Langkah selanjutnya adalah melakukan augmentasi data untuk mendapatkan dataset dengan ukuran yang lebih besar dan lebih variatif.

### D. Hasil Eksperimen

Hasil eksperimen model bisa dilihat dari tiga buah metrik yang digunakan, yaitu Akurasi Prediksi Kategori, Presisi, dan *recall* dari sebuah model. Berikut adalah hasil eksperimen dari setiap model.

Tabel II. Tabel Akurasi Prediksi Kategori tiap Model

Model	Top-1 Accuracy	Top-2 Accuracy	Top-3 Accuracy	Loss
Model Sendiri	0.7160	0.8530	0.9124	1.026
VGG	0.7084	0.8466	0.9083	0.7034
ResNet	0.8640	0.9177	0.9577	0.6230
MobileNet	0.8111	0.8986	0.9418	1.1607
Inception	0.8254	0.9090	0.9452	0.5522

Tabel III. Tabel Presisi tiap model

Kategori	Model Sendiri	VGG	Res Net	Mobile Net	Inception
Kaca mata	0.646	0.640	0.842	0.702	0.771
Baju	0.683	0.668	0.901	0.763	0.838
Celana	0.748	0.706	0.894	0.834	0.892
Dompot	0.835	0.843	0.967	0.935	0.961
Jam tangan	0.695	0.722	0.822	0.762	0.782
Kaos tangan	0.776	0.753	0.898	0.835	0.886
Sepatu	0.704	0.689	0.889	0.820	0.849
Topi	0.825	0.815	0.913	0.883	0.901
Rata-Rata	0.739	0.729	0.891	0.817	0.860

Tabel IV. Tabel Recall tiap Model

Kategori	Model Sendiri	VGG	Res Net	Mobile Net	Inception
Kaca mata	0.378	0.355	0.661	0.554	0.619
Baju	0.126	0.109	0.556	0.143	0.204
Celana	0.231	0.256	0.402	0.304	0.336
Dompot	0.508	0.445	0.760	0.759	0.783
Jam tangan	0.065	0.056	0.331	0.084	0.111
Kaos tangan	0.378	0.398	0.692	0.501	0.593
Sepatu	0.339	0.302	0.655	0.469	0.513
Topi	0.489	0.466	0.562	0.605	0.614
Rata-Rata	0.314	0.298	0.577	0.427	0.472

## VI. EVALUASI DAN ANALISIS

Evaluasi akan dilakukan dengan membandingkan hasil eksperimen dari setiap model, dan nantinya akan dilakukan analisis untuk mengetahui kenapa model memiliki performa seperti itu beserta kelemahan dan kelebihan setiap model. Setelah itu akan dipilih model terbaik yang akan digunakan untuk membangun sistem pencarian gambar.

Nilai akurasi menandakan seberapa besar kebenaran model dalam melakukan prediksi kategori/klasifikasi gambar sesuai dengan kategorinya, sedangkan nilai loss menunjukkan seberapa jauh kesalahan model dalam mengklasifikasikan sebuah gambar.

Untuk mengukur seberapa bagus sistem yang akan dibangun, metrik yang biasanya digunakan untuk mengukur performa dari sebuah sistem adalah metrik precision/presisi dan *recall*. Presisi akan digunakan untuk mengukur seberapa relevan gambar-gambar hasil pencarian yang dikembalikan oleh sistem pencarian gambar, misalnya jika masukan gambar adalah sebuah kacamata maka metrik ini akan mengukur seberapa besar rasio kacamata dalam gambar-gambar hasil pencarian. *Recall* akan digunakan untuk mengukur seberapa besar bagian dari gambar yang dikembalikan sebagai hasil pencarian oleh sistem, misalnya jika pencarian gambar dengan masukan kacamata, maka *recall* menunjukkan rasio banyaknya kacamata dalam gambar-gambar hasil pencarian dengan jumlah total gambar kacamata dalam dataset.

### A. Evaluasi dan Analisis Model Buatan Sendiri

Pada pembuatan model ini, arsitektur yang digunakan adalah *residual network*. Alasan penggunaan arsitektur ini adalah residual network bisa mengatasi permasalahan diminishing gradient sehingga meskipun percobaan nantinya akan memerlukan network yang lebih dalam, model akan bisa mendapatkan hasil yang memuaskan. Model ini memiliki akurasi sekitar 0.7160, 0.8530, 0.9124 untuk top-1, top-2, dan top-3 akurasi. Nilai loss pada model ini adalah 1.026. Nilai presisi dan juga *recall* dari model ini adalah 0.7394 dan 0.3147. Terlihat dari nilai presisi dan bahwa hasil kembalian model ini sudah mayoritas sesuai dengan gambar masukan,

yaitu 73.94% sesuai. Namun model ini hanya bisa mengembalikan sebagian kecil dari semua data dalam kategori tersebut. Kelemahan pada model ini adalah lapisan konvolusi yang digunakan terlalu sedikit sehingga model tidak bisa mendapatkan fitur yang lebih spesifik dari gambar. Alasannya sendiri adalah kurangnya dataset yang disediakan untuk pelatihan model ini, meskipun secara arsitektur model ini bisa dibuat sangat dalam tapi jika memaksakan kedalam arsitektur tanpa diikuti dengan banyaknya data training akan membuat model overfit terhadap data tersebut dan fokus terhadap noise dari gambar dibanding dengan fitur dari gambar sendiri. Kelebihan model ini adalah meskipun memiliki jumlah lapisan sedikit namun bisa mendapatkan nilai akurasi yang cukup bagus.

## B. VGG

VGG adalah sebuah model yang sangat simple dan mudah untuk dimengerti dan menjadi salah satu model yang sering digunakan untuk belajar *machine learning*, ini adalah salah satu kelebihan dari model ini. Kelemahan VGG itu sendiri, yaitu mempunyai terlalu banyak parameter, ukuran model yang sangat besar (parameter sekitar 138 juta), dan juga waktu pelatihan yang lama juga disebabkan oleh kelemahan pada eksperimen ini yaitu jumlah datanya yang sedikit, ini membuat model ini menjadi kurang bisa mempelajari fitur-fitur penting dalam gambar dan mendapatkan hasil yang agak kurang memuaskan, yaitu untuk top-1, top-2, dan top-3 akurasi sekitar 0.7084, 0.8466, 0.9083 dan nilai loss sebesar 0.7034. Sedangkan untuk metrik presisi dan *recall* adalah 0.7299 dan 0.2989. Model ini juga mirip dengan model sebelumnya, meskipun gambar-gambar hasil kembalinya sudah mayoritas sesuai dengan gambar masukan namun hanya bisa mengembalikan sebagian kecil dari data sebenarnya.

## C. Inception

GoogleNet/Inception adalah sebuah model yang berfokus di *computational cost*. Model ini hanya memiliki parameter sekitar 4 juta, sangat kecil jika dibandingkan dengan VGG namun dengan arsitektur yang lebih kompleks. Hal ini dikarenakan pada model ini tidak mengaplikasikan lapisan *Fully-Connected* dan menggantinya dengan lapisan *pooling* saja. Parameter yang lebih sedikit ini mengakibatkan ukuran model yang lebih kecil dan juga proses penghitungan model lebih cepat jika dibandingkan dengan model lain. Hasil dari eksperimen ini sendiri mendapatkan top-1, top-2, dan top-3 akurasi sekitar 0.8254, 0.9090, 0.9452 dengan nilai loss sebesar 0.5522. Akurasi ini sudah dianggap cukup bagus jika dibandingkan dengan akurasi dari model VGG. Presisi dan *recall* dari model ini sendiri mencapai 0.8603 dan 0.4721. Model ini lebih bagus dari model sebelumnya, model VGG, sesuai dengan nilai presisi dan *recall* tersebut, model ini hampir bisa mengembalikan data yang sesuai dengan gambar masukan merepresentasikan hampir setengah data aslinya dan hasil kembalinya sendiri tingkat kesesuaiannya mencapai 86.03%. Kelemahan model ini adalah saat melakukan komputasi diperlukan memori yang cukup banyak dikarenakan model ini fokus pada lebar dari lapisan konvolusi.

## D. ResNet

Model ini berfokus pada *computational accuracy* dan juga merupakan model dengan lapisan paling banyak. Hal ini memungkinkan karena resnet menggunakan *Residual Network* sehingga tidak perlu mengkhawatirkan permasalahan *diminishing gradient* yang membuat performa model menurun jika model terlalu banyak lapisan. Kelebihan model ini tentunya adalah akurasi yang tinggi dibanding dengan model lainnya namun kecepatan training modelnya sendiri juga cepat meskipun tidak secepat GoogleNet/Inception. Kelemahan dari model ini sendiri adalah arsitektur model terlalu fokus pada factor kedalaman model dan tidak mengembangkan factor lebar. Untuk akurasi dari model ini adalah 0.8640, 0.9177, 0.9577 untuk top-1, top-2, dan top-3 sedangkan untuk nilai loss nya sendiri bernilai 0.6230. Nilai presisi dan *recall* yang dihasilkan model ini adalah 0.8912 dan 0.5779. Model ini adalah model dengan hasil presisi dan *recall* yang paling bagus dari semua model yang ada. Model ini dapat mengembalikan data dengan tingkat kesesuaian hampir mencapai 90%, yaitu 89.12% dan satu-satunya model yang bisa mengembalikan hasil pencarian yang merepresentasikan lebih dari setengah data dengan kategori yang sesuai.

## E. MobileNet

MobileNet adalah sebuah model yang didesain untuk bisa dijalankan dalam perangkat mobile dan bisa mempunyai performa yang bisa diterima di perangkat tersebut. Model ini adalah sebuah model yang sangat ringan jika dibandingkan dengan model-model sebelumnya, terlihat ukuran model ini hanya sekitar 1/5 – 1/2 dari model lainnya. Model ini juga didesain untuk melakukan proses komputasi dengan cepat. Meskipun begitu model ini memiliki kelemahan akibat mengedepankan kecepatan dan juga keringanan model itu sendiri, yaitu nilai loss dari model ini adalah yang tertinggi yaitu 1.1607. Nilai ini berarti prediksi model untuk data test adalah paling jauh dibanding dengan model lainnya. Nilai akurasi dari model ini sendiri cukup bagus yaitu 0.8111, 0.8986, 0.9418. Untuk nilai presisi dan *recall* pada model ini bernilai 0.8171 dan 0.4278. Model ini memiliki nilai presisi dan *recall* yang lebih buruk jika dibandingkan 2 model sebelumnya, GoogleNet/Inception dan Resnet, tapi lebih tinggi dari 2 model sisanya.

## F. Perbandingan Akhir Model

Setiap model diatas memiliki kelebihan dan kekurangannya sendiri, misalnya model paling ringan adalah MobileNet, model paling cepat dan efisien adalah GoogleNet/Inception, dan model dengan akurasi, presisi, dan *recall* terbaik adalah Resnet. Kekurangan dari model-model tersebut juga bervariasi, mulai dari memori, nilai loss, *recall*, lamanya proses pelatihan, besarnya model yang dihasilkan. Dari Kelebihan dan kelemahan tersebut model yang akan dipilih untuk digunakan dalam sistem adalah model Resnet, alasannya adalah model ini memiliki nilai akurasi paling tinggi, 86.40%, presisi yang paling tinggi, 89.12%, nilai *recall* tertinggi 57.79%. Model pencarian gambar yang ideal adalah model yang bisa mengembalikan gambar yang sesuai/relevan dengan gambar masukan, ini ditampilkan

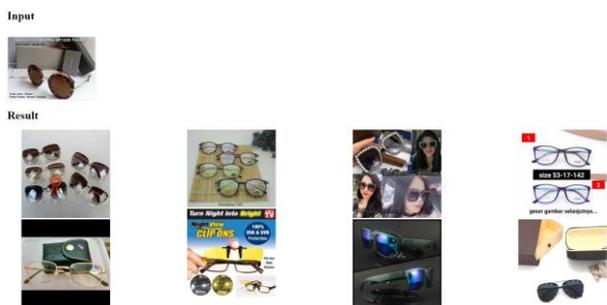
dengan nilai presisi dari model. Model juga diharapkan bisa memiliki nilai *recall* yang tidak terlalu rendah dan masih bisa merepresentasikan data asli setiap kategori. Selain itu model Resnet tidak memiliki kekurangan fatal dibanding dengan model lain, baik dari segi ukuran model, lama pelatihan, akurasi, dan nilai loss dari model sendiri.

### VII. PEMBANGUNAN SISTEM PENCARIAN GAMBAR

Pembangunan sistem pencarian gambar akan dilakukan dengan mengambil model terbaik hasil eksperimen. Ekspor model dilakukan dengan bantuan library H5py yang akan menyimpan model yang sudah dilatih tersebut dan juga dataset pelatihan juga ekstraksi fitur beserta labelnya. Data tersebut nantinya diperlukan untuk pembuatan sistem pencarian gambar. Setelah data dan model disimpan akan dibuat sebuah sistem berbasis web dengan bantuan aplikasi flask dan menggunakan Bahasa python. Interface sistem hanya terdapat input masukan dan juga 8 hasil pencarian gambar. Berikut adalah contoh *interface* sistem:



Gambar IV Interface Awal Sistem Pencarian Gambar



Gambar V Interface Hasil Pencarian Gambar

### VIII. KESIMPULAN

Arsitektur terbaik pada eksperimen pembangunan sistem ini adalah sistem yang menggunakan metode transfer learning dengan model ResNet. Model ini mendapatkan nilai akurasi prediksi kategori sebesar 0.8640, nilai presisi sebesar 0.8912, dan nilai recall sebesar 0.5779..

Modul-modul yang bisa meningkatkan perfoma sebuah sistem pencarian gambar selain modul utama yaitu modul ekstraksi fitur dan klasifikasi adalah modul augmentasi data

dan juga modul *reindexing*. Modul augmentasi data sangat membantu terutama jika dataset yang dimiliki berukuran kecil sedangkan modul *reindexing* akan sangat membantu dalam proses pencarian gambar dan membandingkan fitur antara fitur data query dan fitur data di database.

Pelatihan model dari awal memerlukan dataset yang besar (sekitar 10-25k data per kelas) agar bisa menghasilkan model yang bagus. Sedangkan *transfer learning* membutuhkan dataset yang lebih kecil jika dibandingkan dengan pelatihan model dari awal. Hasil *Transfer learning* cenderung lebih bagus daripada pembuatan model dari awal, terutama semakin kecil dataset yang digunakan.

Pengembangan selanjutnya yang bisa dilakukan untuk meningkatkan performa sistem ini adalah melakukan *parallelism* terhadap model, sehingga bisa mengekstraksi fitur dari dataset secara bersamaan dan mempercepat pembuatan database maupun penambahan data terhadap database sendiri.

### REFERENCES

- [1] Yushi Jing, David Liu, Dmitry Kislyuk, Andrew Zhai, Jiaping Xu, Jeff Donahue, Sarah Tavel. *Visual search at Pinterest*. University of California, Berkeley and Visual Discovery, Pinterest (2015).
- [2] Yanhao Zhang, Pan Pan, Yun Zheng, Kang Zhao, Yingya Zhang, Xiaofeng Ren, Rong Jin. *Visual search at Alibaba*. Machine Intelligence Technology Lab, Alibaba Group(2018).
- [3] Houdong Hu, Yan Wang, Linjun Yang, Pavel Komlev, Li Huang, Xi (Stephen) Chen, Jiawei Huang, Ye Wu, Meenaz Merchant, Arun Sacheti. *Web-Scale Responsive Visual search at Bing* (2018).
- [4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. *Imagenet classification with deep Convolutional Neural Networks*. In Advances in neural information processing systems(NIPS). 1097–1105.
- [5] Joe Yue-Hei Ng, Fan Yang, and Larry S. Davis. 2015. *Exploiting local features from deep networks for image retrieval*. In IEEE Conference on Computer vision and Pattern Recognition Workshops, CVPR Workshops. 53–61.
- [6] Balabanovic, M. (1998). *Learning to surf: Multi-agent systems for adaptive web page recommendation*. Doctoral dissertation, Stanford University, Menlo Park, CA: Department of Computer Science.
- [7] McKusick, K.B., & Langley, P. (1991). Constraints on tree structure in concept formation. *Prosiding The 21st ACM-SIGIR International Conference on Research and Development in Information Retrieval*, 206-214. New York, NY:ACM Press.
- [8] K. Simonyan and A. Zisserman. Very deep Convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.
- [9] A. Krizhevsky, S. Ilya, and G. E. Hinton. *Imagenet classification with deep Convolutional Neural Networks*. In Advances in Neural Information Processing Systems (NIPS), pages 1097{1105. 2012.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. *Deep Residual Learning for Image Recognition*. In IEEE Conference on Computer vision and Pattern Recognition(CVPR). 770–778.
- [11] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. 2015. *Going deeper with convolutions*. In IEEE Conference on Computer vision and Pattern Recognition(CVPR). 1–9.