

Deteksi Kerumunan Menggunakan Metode *Fully-Convolutional Network* pada Kamera

Muhammad Fatih, Nanik Suciati dan Dini Adni Navastara
Departemen Teknik Informatika, Institut Teknologi Sepuluh Nopember (ITS)
e-mail: nanik@if.its.ac.id

Abstrak—Pada masa pandemi virus COVID-19 pemerintah menetapkan peraturan yang mengharuskan masyarakat untuk menerapkan beberapa protokol kesehatan. Salah satunya adalah menghindari kerumunan dan menjaga jarak. Untuk membantu pengawasan kepatuhan masyarakat terhadap protokol tersebut pada area yang luas, diperlukan sebuah sistem monitoring untuk memantau adanya kerumunan dengan menggunakan drone. Video yang direkam menggunakan kamera drone diproses menggunakan metode *Fully-Convolutional Network* (FCN) dengan menggabungkan *loss function* untuk tugas klasifikasi yang menentukan kerumunan atau tidak dan *loss function* untuk tugas *regression* yang menghitung kepadatan berdasarkan rata-rata *clustering coefficient*. Penelitian ini mengimplementasikan metode FCN dengan input berupa rangkaian gambar yang diambil dari video sehingga menghasilkan output berupa keputusan apakah sejumlah orang dalam gambar itu berkerumun atau tidak. Data latih yang digunakan adalah *VisDrone Dataset* dan *P-DESTRE Dataset* yang terdiri dari rangkaian gambar yang direkam menggunakan drone yang diterbangkan dengan ketinggian rata-rata dengan mengambil contoh video berisi kerumunan dan bukan kerumunan. Hasil pengujian terbaik didapatkan menggunakan *pre-trained model 5* dimana memiliki 2 keluaran yaitu 1 klasifikasi dan 1 regresi yaitu memiliki akurasi klasifikasi sebesar 0,978 sedangkan *mean absolute error* untuk regresinya sebesar 0,141.

Kata kunci—COVID-19, Crowd Detection, *Fully-Convolutional Network*, *Convolutional Neural Network*, *Clustering Coefficient*, dan *Drone*.

I. PENDAHULUAN

DALAM masa pandemi virus COVID-19 seperti saat perencanaan dan pelaksanaan penelitian ini membuat pemerintah menetapkan peraturan yang mengharuskan masyarakat untuk menerapkan beberapa protokol kesehatan salah satunya adalah dengan menghindari kerumunan dan menjaga jarak. Meskipun peraturan sudah berlaku, masyarakat tetap banyak sekali yang membandel dan tetap berkerumun seperti tidak terjadi apa-apa, oleh karena itu diperlukan sebuah sistem monitoring untuk memantau aksi kerumunan salah satunya dengan menggunakan drone. Drone digunakan karena bisa beroperasi tanpa harus ada orang yang mengendarai atau dikendalikan dengan *remote control* dan bisa mencakup wilayah yang cukup luas sehingga pengawasan dapat dilakukan dengan lebih mudah dengan mengambil video yang direkam menggunakan kamera drone. Video yang direkam menggunakan kamera drone akan diproses dengan menggunakan metode FCN (*Fully-Convolutional Network*) yang mengkonjugasikan komputasi dan efektivitas serta ditambah dengan *two-loss model* yaitu *Classification* untuk menentukan itu kerumunan atau tidak dan *Regression* untuk komputasi jarak antar orang.

Selain metode di atas terdapat juga metode lain untuk menghasilkan hal yang sama diantaranya adalah dimana



Gambar 1. Contoh Dataset *VisDrone*.

penulis menggunakan metode *Convolutional Neural Network* dengan kombinasi *Deep Neural Network* untuk *crowd counting* dan *Shallow Neural Network* untuk mengatasi orang yang sangat jauh dari kamera [1]. Di paper penulis menggunakan metode yang mirip dengan apa yang akan saya gunakan yaitu FCN juga tetapi penulis dalam paper menggunakan model FCN yang berasal dari *pre-trained CNN* yang *fully-connected layer*-nya dibuang dan sebuah *final convolutional layer* ditambahkan sebelum *re-training* semua *convolutional layer*-nya [2]. Dalam paper penulis bermaksud meningkatkan performa paper mereka sebelumnya yaitu paper dengan menggunakan *regularization scheme* yang terinspirasi dari *Graph Embedded Convolutional Neural Network* [2-3].

Penelitian ini mengimplementasikan metode FCN (*Fully-Convolutional Network*) dalam citra gambar sesuai dengan *paper* referensi sehingga menghasilkan output berupa keputusan apakah sejumlah orang itu kerumunan atau tidak beserta dengan *heatmap*-nya sehingga diharapkan dapat dilakukan tindakan dari pihak berwenang untuk mengurangi tingkat penyebaran virus COVID-19 ini di Indonesia [4].

II. DASAR TEORI

Bagian ini menjelaskan dasar teori berbagai metode yang akan digunakan kedepannya.

A. Dataset *VisDrone*

VisDrone dataset berisi kumpulan foto orang berkumpul baik sedikit atau banyak yang diambil menggunakan *drone* [4]. Dataset *P-DESTRE* yang digunakan penulis terdiri dari 6 video orang berkumpul dimana setiap video diekstrak tiap *frame* menjadi bentuk gambar sehingga hasil datanya adalah gambar per *frame* sama seperti *VisDrone* di atas. Jumlah tiap *frame* bervariasi dari 500-1.500 *frame* sehingga total data gambar yang digunakan adalah 6.521 data gambar. Contoh dataset bisa dilihat pada Gambar 1 [4].



Gambar 2. Contoh Dataset P-DESTRE. (kiri) Portugal, (kanan) India.

B. Dataset P-DESTRE

Dataset P-DESTRE awalnya merupakan data yang digunakan untuk keperluan *pedestrian identification* dari data *drone* [5]. Dataset ini merupakan data video yang masih berupa video di daerah University of Beira Interior di Portugal dan JSS Science and Technology University di India dimana berisi kumpulan data video orang berkumpul baik banyak maupun sedikit yang diambil menggunakan *drone*. Data terdiri dari satu *folder* berisi Video dan satu *folder* berisi kumpulan anotasi dalam bentuk txt. Contoh dataset bisa dilihat pada Gambar 2 [5].

C. CNN (Convolutional Neural Network)

CNN adalah salah satu jenis *neural network* yang biasa digunakan pada data *image*. CNN bisa digunakan untuk mendeteksi dan mengenali *object* pada sebuah *image*. CNN memanfaatkan proses konvolusi dengan menggerakkan sebuah kernel konvolusi (*filter*) berukuran tertentu ke sebuah gambar, komputer mendapatkan informasi representatif baru dari hasil perkalian bagian gambar tersebut dengan *filter* yang digunakan. Arsitektur dari CNN dibagi menjadi 2 bagian besar, *Feature Extraction Layer* dan *Fully-Connected Layer* (MLP) [6].

D. FCN (Fully-Convolutional Network)

FCN adalah arsitektur yang biasanya digunakan untuk *semantic segmentation* menggunakan *locally connected layers*, seperti *convolution*, *pooling*, dan *upsampling*. FCN tidak berisi *dense layer* seperti pada CNN pada umumnya melainkan berisi konvolusi 1x1 yang menjalankan tugas *fully connected layers*. Menghindari penggunaan *dense layer* berarti lebih sedikit parameter dimana membuat jaringan lebih cepat untuk di-*train*. FCN dapat berfungsi untuk ukuran gambar bervariasi karena semua koneksi bersifat local.

E. Euclidean Distance

Jarak *Euclidean* dalam ruang *Euclidean* merupakan jarak antar dua titik. Mirip dengan *Pythagoras* dalam bidang kartesian, jarak *Euclidean* dapat menghitung jarak meskipun dalam dimensi yang lebih dari 2 (n-space) [7].

F. Binary Cross-entropy Loss Function (untuk klasifikasi)

Persamaan 1 mengukur kinerja model klasifikasi yang *output* nya berupa nilai probabilitas antara 0 dan 1. Persamaan 1 berikut merupakan rumus dari *Cross-entropy loss Function*.

$$H(\theta) = \sum_{i=1}^N y_i^c \log(h_{\theta}^c(x_i)) + (1 - y_i^c) \log(1 - h_{\theta}^c(x_i)) \quad (1)$$

Dimana: N = *Sample cardinality*
 y_i = label kelas asli
 $h_{\theta}(x_i)$ = label kelas prediksi

c = *classification task*
 θ = *weight parameters*
 x_i = *single training sample*

G. Mean Absolute Error Loss Function (untuk regresi)

Persamaan 2 berfungsi untuk mendukung *Classification task* diatas sebagai *auxiliary output* dimana *loss function* ini adalah rata rata dari *absolute error* yang melibatkan jumlah orang yang sebenarnya dan yang diprediksi. Persamaan 2 merupakan rumus dari *Loss Function* untuk regresi.

$$L(\theta) = \frac{1}{N} \sum_{i=1}^N |y_i^r - h_{\theta}^r(x_i)| \quad (2)$$

Dimana: N = *Sample cardinality*
 y_i = label kelas asli
 $h_{\theta}(x_i)$ = label kelas prediksi
 r = *regression task*
 θ = *weight parameters*

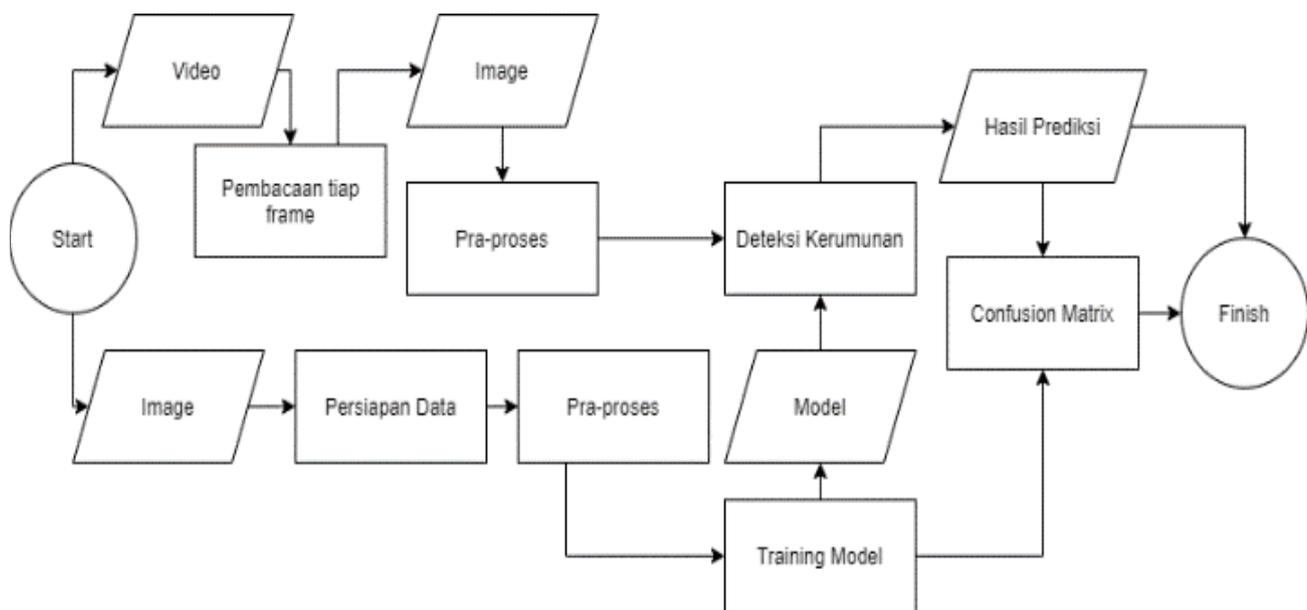
H. Confusion Matrix

Salah satu teknik aplikasi pada *machine learning* adalah *supervised learning*. Klasifikasi merupakan *supervised learning*, yang merupakan model prediksi dimana hasil prediksinya bersifat diskrit. Cara mengukur performa dari model klasifikasi yang digunakan adalah membandingkan nilai aktual dengan nilai prediksi. *Confusion Matrix* adalah pengukuran performa untuk masalah klasifikasi *machine learning* dimana keluaran dapat berupa dua kelas atau lebih. *Confusion Matrix* adalah tabel dengan 4 kombinasi berbeda dari nilai prediksi dan nilai aktual. Ada empat istilah yang merupakan representasi hasil proses klasifikasi pada *confusion matrix* yaitu *True Positif*, *True Negatif*, *False Positif*, dan *False Negatif*.

Nilai Prediksi adalah keluaran dari program dimana nilainya Positif dan Negatif. Nilai Aktual adalah nilai sebenarnya dimana nilainya *True* dan *False*. Dari kesimpulan diatas dapat dihitung nilai *accuracy*, *precision*, *recall* dan *FIScore*. *Accuracy* menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar. *Precision* menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model. *Recall* menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi. *F-IScore* menggambarkan perbandingan rata-rata *precision* dan *recall* yang dibobotkan. *Accuracy* tepat kita gunakan sebagai acuan performansi algoritma jika dataset kita memiliki jumlah data *False Negative* dan *False Positive* yang sangat mendekati (*symmetric*). Namun jika jumlahnya tidak mendekati, maka sebaiknya kita menggunakan *F-IScore* sebagai acuan.

I. Clustering Coefficient

Dalam teori grafik, *clustering coefficient* adalah ukuran sejauh mana *node* dalam grafik cenderung mengelompok bersama. Bukti menunjukkan bahwa di sebagian besar *network* dunia nyata, dan di *social network* tertentu, node cenderung membuat grup yang terjalin erat yang ditandai dengan kepadatan ikatan yang relatif tinggi, kemungkinan ini cenderung lebih besar dari probabilitas rata-rata hubungan yang dibuat secara acak antara dua node (*Holland dan Leinhardt, 1971; Watts dan Strogatz, 1998*). Ada dua versi ukuran ini: global dan lokal. Versi global dirancang untuk



Gambar 3. Diagram Alir Gambaran Arsitektur Secara Umum.

memberikan indikasi keseluruhan dari pengelompokan di sebuah *network*, sedangkan lokal memberikan indikasi keterikatan *single node*.

III. DESAIN

Bagian ini menjelaskan secara garis besar tentang program yang dapat mendeteksi apakah tiap sekian *frame* yang ada dalam video merupakan kerumunan orang atau bukan menggunakan metode *Fully-Convolutional Network* (FCN). Pengerjaan penelitian ini berdasarkan sumber referensi yang memiliki 3 proses utama yaitu *Dataset Preparation*, *Training Model*, dan *Testing Model*. Untuk *flowchart* nya bisa dilihat pada Gambar 3.

A. Persiapan Dataset

Pertama-tama, Penulis melakukan persiapan data pada data latih yang sama dengan *paper* yaitu Dataset Visdrone dan menggunakan data lain yang bernama Dataset P-DESTRE [4].

1) Dataset VisDrone

Dataset VisDrone yang digunakan penulis terdiri dari 80 *sequence* gambar dimana setiap *sequence* terdiri dari 300-1000 gambar per *frame* sehingga data gambar yang digunakan total dari Dataset VisDrone ini adalah 23.541 data gambar. Penulis membedakan data gambar yang tersedia menjadi 2 kelas yaitu *crowd* dan *non crowd* menggunakan parameter dari *paper* dan menggunakan pengamatan penulis sendiri. Parameter yang diberikan *paper* adalah 10 orang di tiap gambar atau lebih maka dianggap masuk kelas *crowd*, sedangkan dari pengamatan penulis adalah jika ada 6 atau lebih orang yang berkumpul dan berjarak dekat dalam tiap gambar maka dianggap masuk kelas *crowd*.

2) Dataset P-DESTRE

Dataset P-DESTRE yang digunakan penulis terdiri dari 6 video orang berkumpul dimana setiap video diekstrak tiap *frame* menjadi bentuk gambar sehingga hasil datanya adalah gambar per *frame* sama seperti VisDrone diatas. Jumlah tiap *frame* bervariasi dari 500-1.500 *frame* sehingga total data gambar yang digunakan adalah 6.521 data gambar.

3) Dataset Gabungan

Karena pada dataset VisDrone penulis kekurangan data untuk kelas *crowd* saja maka untuk dataset P-DESTRE ini penulis hanya menggunakan untuk kelas *crowd* saja dan memilih secara manual dari pengamatan penulis seperti data VisDrone tadi dimana pengamatan penulis adalah jika ada 6 atau lebih orang yang berkumpul atau berjarak dekat dalam tiap gambar maka dianggap masuk kelas *crowd*. Jika hanya menggunakan data VisDrone saja maka persebaran data per kelasnya adalah pada Tabel 1.

Jika dengan menambah data P-DESTRE persebaran data per kelasnya adalah pada Tabel 2.

Selanjutnya dataset Visdrone dimana datanya imbalance akan disebut sebagai data 1 sedangkan dataset VisDrone dengan tambahan P-DESTRE akan disebut sebagai data 2.

4) Persiapan Ground Truth

Pada saat akan membagi data menjadi dua kelas yaitu kelas *crowd* dan *non crowd*, pada tiap dataset yaitu VisDrone dan P-DESTRE, setiap *frame* juga dihitung jumlah orang yang ada dalam gambar dalam *frame* tersebut dan rata-rata *clustering coefficient* dan disimpan ke dalam csv sehingga data yang ada didalam csv adalah *path image*, kelas *crowd* atau *non crowd*, *clustering coefficient* (rata-rata), dan jumlah orang (*people count*) dimana yang tersedia dalam anotasi dataset adalah *bounding box* tiap orang sehingga penulis harus menghitung kedua hal diatas. Untuk *clustering coefficient* penulis menggunakan *graph* dimana setiap orang diwakilkan di tengah *bounding box* mereka sebagai satu titik dan jarak antar orang dihitung per *pixel* dan untuk apakah dua titik itu masuk dalam *cluster* yang sama, penulis menggunakan parameter dari *paper* yaitu 1/10 lebar gambar dengan kombinasi manual dari penulis karena setiap gambar yang tersedia tidak ditangkap dari ketinggian yang sama sehingga ada kesalahan hasil jika semua disama ratakan. Untuk *people count* penulis hanya tinggal menghitung jumlah titik atau *node* dari *graph* yang sudah terbentuk tadi. Data juga dibagi dalam tiga kategori yaitu *train*, *test*, dan *validation* dimana penulis membagi dengan cara membagi tiap *sequence* gambar dengan 80% pertama masuk *train*, 10%

Tabel 1.
Persebaran Dataset VisDrone

	Crowd	Non Crowd
Train	6.854	11.912
Val	810	1.585
Test	846	1.534

Tabel 2.
Persebaran Dataset VisDrone + P-DESTRE

	Crowd	Non Crowd
Train	11.912	11.912
Val	1.585	1.585
Test	1.534	1.534

selanjutnya masuk *test*, dan sisanya yaitu 10% lagi masuk *validation*.

B. Training Model

Pada bagian ini setelah data sudah dipersiapkan dan sudah terbagi menjadi 2 kelas yaitu *crowd* dan *non crowd* serta *csv ground truth* untuk *training* siap maka data sudah bisa untuk dilakukan *train* menggunakan model, penulis mengambil referensi dari *paper* utama dimana mengambil model yang sama dengan *paper* utama dengan sedikit modifikasi yaitu pada bagian *learning rate* penulis menggunakan 0,0001 sedangkan *paper* menggunakan 0,1 tujuannya adalah agar performa meningkat dan model dapat belajar dengan lebih baik. Penulis juga mengganti parameter fungsi aktivasi dimana referensi utama tidak menggunakan fungsi aktivasi pada layer output regresi menjadi menggunakan fungsi aktivasi *relu* tujuannya adalah agar hasil regresi tidak kurang dari 0. Penulis menggunakan enam model yaitu satu model untuk satu klasifikasi biner, empat model untuk satu klasifikasi dan satu regresi, dan satu model untuk satu klasifikasi dan dua regresi. Regresi disini bertujuan untuk membantu prediksi dari klasifikasi yang menjadi tujuan utamanya, meskipun tidak menutup kemungkinan juga terjadi penurunan performa, tetapi diharapkan akan terjadi kenaikan performa dengan adanya regresi ini. Empat model pertama merupakan model dari *scratch* yang dibuat penulis berdasarkan referensi *paper* sedangkan 2 sisanya merupakan model *pre-trained* dari referensi *paper*.

1) Model 1 (*crowd*)

Model 1 ini ditujukan untuk tujuan klasifikasi saja dimana hanya ada satu keluaran model yaitu kelas *crowd* dan kelas *non crowd* yang dilambangkan dengan *float* dari skala 0-1 dimana jika hasil *float* kurang dari 0,5 akan masuk ke kelas *non crowd* dan jika hasil *float* 0,5 atau lebih akan masuk ke kelas *crowd*.

2) Model 2 (*crowd + coeff*)

Model 2 ini ditujukan untuk klasifikasi dan regresi dimana regresi bertujuan untuk membantu proses klasifikasi, model ini mempunyai dua keluaran yaitu klasifikasi dan regresi dimana klasifikasinya yaitu kelas *crowd* dan kelas *non crowd* yang dilambangkan dengan *float* dari skala 0-1 dimana jika hasil *float* kurang dari 0,5 akan masuk ke kelas *non crowd* dan jika hasil *float* 0,5 atau lebih akan masuk ke kelas *crowd*, sedangkan untuk regresinya yaitu berupa *float* dengan skala 0-1 yang melambangkan rata rata *clustering coefficient*.

3) Model 3 (*crowd + count*)

Model 3 ini bertujuan untuk klasifikasi dan regresi dimana regresi bertujuan untuk membantu proses klasifikasi, model

Tabel 3.
Hasil Validasi dengan Data 1

Model	MAE		Loss	Accuracy
	Coeff	Count		
1	-	-	0,324	0,868
2	0,189	-	0,598	0,827
3	-	10,161	10,663	0,853
4	0,276	10,291	11,06	0,845
5	0,169	-	0,344	0,939
6	-	10,371	10,815	0,843

ini mempunyai dua keluaran yaitu klasifikasi dan regresi dimana klasifikasinya yaitu kelas *crowd* dan kelas *non crowd* yang dilambangkan dengan *float* dari skala 0-1 dimana jika hasil *float* kurang dari 0,5 akan masuk ke kelas *non crowd* dan jika hasil *float* 0,5 atau lebih akan masuk ke kelas *crowd*, sedangkan untuk regresinya yaitu berupa *float* dengan skala 0-∞ yang melambangkan jumlah orang atau *people count*.

4) Model 4 (*crowd + coeff + count*)

Model 4 ini bertujuan untuk klasifikasi dan regresi dimana regresi bertujuan untuk membantu proses klasifikasi, model ini mempunyai tiga keluaran yaitu klasifikasi dan regresi dimana klasifikasinya yaitu kelas *crowd* dan kelas *non crowd* yang dilambangkan dengan *float* dari skala 0-1 dimana jika hasil *float* kurang dari 0,5 akan masuk ke kelas *non crowd* dan jika hasil *float* 0,5 atau lebih akan masuk ke kelas *crowd*, sedangkan untuk regresinya yaitu berupa *float* dengan skala 0-1 yang melambangkan rata rata *clustering coefficient* dan berupa *float* juga dengan skala 0-∞ yang melambangkan jumlah orang atau *people count*.

5) Model 5 (*crowd + coeff*)

Model 5 ini merupakan model *pre-trained* yang bertujuan untuk klasifikasi dan regresi dimana regresi bertujuan untuk membantu proses klasifikasi, model ini mempunyai dua keluaran yaitu klasifikasi dan regresi dimana klasifikasinya yaitu kelas *crowd* dan kelas *non crowd* yang dilambangkan dengan *float* dari skala 0-1 dimana jika hasil *float* kurang dari 0,5 akan masuk ke kelas *non crowd* dan jika hasil *float* 0,5 atau lebih akan masuk ke kelas *crowd*, sedangkan untuk regresinya yaitu berupa *float* dengan skala 0-1 yang melambangkan rata rata *clustering coefficient*.

6) Model 6 (*crowd + count*)

Model 6 ini bertujuan untuk klasifikasi dan regresi dimana regresi bertujuan untuk membantu proses klasifikasi, model ini mempunyai dua keluaran yaitu klasifikasi dan regresi dimana klasifikasinya yaitu kelas *crowd* dan kelas *non crowd* yang dilambangkan dengan *float* dari skala 0-1 dimana jika hasil *float* kurang dari 0,5 akan masuk ke kelas *non crowd* dan jika hasil *float* 0,5 atau lebih akan masuk ke kelas *crowd*, sedangkan untuk regresinya yaitu berupa *float* dengan skala 0-∞ yang melambangkan jumlah orang atau *people count*.

7) Proses Training

Pada proses ini model akan ditrain dengan melalui pra-proses terlebih dahulu yaitu dengan *resize* ke 128x128 sesuai dengan *input layer* modelnya sekaligus menunjuk ke data *csv* dan *folder* gambar dengan menggunakan metode dari keras yaitu *flow from directory* untuk model 1 dan *flow from dataframe* untuk sisanya dengan parameter *shuffle* aktif di *train* dan *validation* namun tidak untuk *test* sedangkan untuk *class mode* menggunakan *multi output*. Dalam 50 *epoch* training penulis menggunakan *early stop* dengan

Tabel 4.
Hasil validasi dengan data 2

Model	MAE		Loss	Accuracy
	Coeff	Count		
1	-	-	0,47	0,770
2	0,209	-	0,503	0,890
3	-	9,826	10,099	0,892
4	0,239	8,717	9,208	0,894
5	0,203	-	0,341	0,928
6	-	8,729	9,033	0,910

Tabel 5.
Hasil Uji Coba Bobot

Model	Weight	Data	Accuracy
1	Yes	1	0,836
1	No	1	0,772
1	No	2	0,868

memperhatikan peningkatan *val_loss* dalam kurun 7 *epoch* dan menyimpan model versi terbaik dilihat dari *val_loss* terkecil. Proses *training* juga menggunakan *library* untuk mengurangi *learning rate* jika tidak ada perkembangan dalam proses *training* yaitu dengan menggunakan *ReduceLRonPlateau*. Training dilakukan per *batch* dengan ukuran *batch* 64.

C. Testing Model

Testing dilakukan dengan menggunakan cara yang hampir sama seperti training yaitu *resize* ke 128x128 dan menunjuk ke data csv dan *folder* gambar dengan menggunakan metode dari keras yaitu *flow from directory* untuk model 1 dan *flow from dataframe* untuk model 2, 3, dan 4 dimana sebelumnya pada persiapan data sudah di pisah untuk *testing*, dilakukan dengan tanpa *batch* sehingga menghasilkan jumlah prediksi yang sama dengan *ground truth*.

Performa diukur dengan menggunakan *accuracy* untuk klasifikasi dengan tampilan *confusion matrix* dan menggunakan *mean absolute error* untuk setiap regresi yang digunakan. Pada *confusion matrix* akan ditampilkan hasil *accuracy*, *precision*, *recall*, dan *f1score* dimana semakin besar angka akan semakin baik performanya, sedangkan untuk *mean absolute error* akan tampil dalam bentuk *float* dimana semakin kecil *float* akan semakin baik performanya.

IV. IMPLEMENTASI

Proses uji coba berguna untuk menemukan skenario yang menghasilkan performa model yang paling optimal. Ada 4 macam skenario uji coba yang akan dicoba pada arsitektur yang telah dirancang. Skenario uji coba yang akan dilakukan yaitu:

1. Hasil Pelatihan.
2. Uji Coba dengan Bobot.
3. Uji Coba Dataset VisDrone.
4. Uji Coba Dataset VisDrone + P-DESTRE.

V. UJI COBA DAN EVALUASI

A. Hasil Pelatihan

Pelatihan dilakukan dengan membentuk dua belas model, dengan 4 model dibuat dari *scratch* dan 2 model *pre-trained* dari *paper*. Pelatihan model dilakukan sampai nilai *loss* tidak mengalami penurunan selama 7 *epoch*. Hasil pelatihan berupa *accuracy* dan *loss* pada *validation set* pada data 1 dapat dilihat pada Tabel 3.

Tabel 6.
Hasil uji coba pada data 1

Model	MAE		Accuracy	F1Score
	Coeff	Count		
1	-	-	0,772	0,835
2	0,167	-	0,822	0,719
3	-	7,430	0,924	0,890
4	0,270	7,330	0,905	0,867
5	0,182	-	0,949	0,949
6	-	8,840	0,875	0,873

Tabel 7.
Hasil uji coba pada data 2

Model	MAE		Accuracy	F1Score
	Coeff	Count		
1	-	-	0,868	0,873
2	0,164	-	0,915	0,914
3	-	7,64	0,921	0,922
4	0,23	7,21	0,945	0,945
5	0,141	-	0,978	0,969
6	-	7,570	0,875	0,873

Hasil pelatihan berupa *accuracy* dan *loss* pada *set* pada data 2 dapat dilihat pada Tabel 4.

Dapat dilihat pada keseluruhan bahwa pada data 2 mengalami kenaikan performa sesuai dengan yang diharapkan.

B. Hasil Uji Coba dengan Bobot

Pada Tabel 5 adalah hasil dari model 1 jika ditest dengan dan tanpa bobot pada data 1 serta pada data 2 tanpa bobot.

C. Hasil Uji Coba pada data 1

Uji coba pada dataset VisDrone dilakukan dengan cara melakukan pengenalan pada seluruh gambar dari data *testing*. Hasil prediksi akan dibandingkan dengan *ground truth* yang sudah tersedia. Penilaian akan dilakukan dengan menggunakan *accuracy*, *F1 score*, dan *mean absolute error*. Hasil uji coba dapat dilihat pada Tabel 6.

D. Hasil Uji Coba pada data 2

Uji coba pada dataset VisDrone + P-DESTRE dilakukan dengan cara melakukan pengenalan pada seluruh gambar dari data *testing*. Hasil prediksi akan dibandingkan dengan *ground truth* yang sudah tersedia. Penilaian akan dilakukan dengan menggunakan *accuracy*, *F1score*, dan *mean absolute error*. Hasil uji coba dapat dilihat pada Tabel 7.

VI. KESIMPULAN DAN SARAN

A. Kesimpulan

Dalam pengerjaan penelitian ini setelah melalui tahap perancangan aplikasi, implementasi metode, serta uji coba, diperoleh kesimpulan sebagai berikut: (1) Pra-proses dataset dilakukan dengan melakukan pengamatan terhadap sampel gambar dalam setiap *sequence* berdasarkan cara mendeteksi kerumunan pada poin 2 sehingga dataset terbagi menjadi 2 kelas dengan jumlah yang seimbang. (2) Cara mendeteksi kerumunan adalah minimal terdapat 6 orang yang berdekatan dalam suatu gambar maka gambar itu masuk ke dalam kelas orang yang berkerumun atau *crowd*. (3) Cara mengukur performa model yang digunakan adalah berdasarkan hasil pengujian yang telah dilakukan. (4) Model dengan performa terbaik pada ketiga uji coba adalah model *pre-trained crowd coeff* yaitu model 5 dengan akurasi untuk klasifikasi sebesar 0,978 dan *mean absolute error* untuk regresi sebesar 0,141.

B. Saran

Saran yang diberikan untuk pengembangan sistem deteksi kerumunan pada data video atau gambar, yaitu: (1) Model di latih dengan penambahan *epoch* diatas 50. (2) Menggunakan parameter *loss function mean squared error* untuk keluaran regresi. (3) Menambah data latih dan data uji agar lebih bervariasi sehingga performa diharapkan lebih baik.

DAFTAR PUSTAKA

- [1] S. Haque, M. S. Sadi, M. E. H. Rafi, M. M. Islam, and M. K. Hasan, "Real-Time Crowd Detection to Prevent Stampede," in *Proceedings of International Joint Conference on Computational Intelligence*, 2020, pp. 665–678.
- [2] M. Tzelepi and A. Tefas, "Human Crowd Detection for Drone Flight Safety using Convolutional Neural Networks," in *2017 25th European Signal Processing Conference (EUSIPCO)*, 2017, pp. 743–747.
- [3] M. Tzelepi and A. Tefas, "Graph embedded convolutional neural networks in human crowd detection for drone flight safety," *IEEE Trans. Emerg. Top. Comput. Intell.*, vol. 5, no. 2, pp. 191–204, 2019.
- [4] G. Castellano, C. Castiello, C. Mencar, and G. Vessio, "Crowd detection in aerial images using spatial graphs and fully-convolutional neural networks," *IEEE Access*, vol. 8, pp. 64534–64544, 2020.
- [5] P. Zhu, L. Wen, D. Du, X. Bian, Q. Hu, and H. Ling, "Vision meets drones: past, present and future," *arXiv Prepr. arXiv2001.06303*, 2020.
- [6] S. V. A. Kumar, E. Yaghoubi, A. Das, B. S. Harish, and H. Proença, "The p-destre: A fully annotated dataset for pedestrian detection, tracking, and short/long-term re-identification from aerial devices," *IEEE Trans. Inf. Forensics Secur.*, vol. 16, pp. 1696–1708, 2020.
- [7] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431–3440.