

Demo Day Final Report

Machine Learning with TensorFlow Training
Professional Academy Digital Talent Scholarship 2022

Group Number	Group 4
Dataset	Handwriting Recognition
Name - DTS ID	<div><div>1. Adi Kurniawan - 152236035100-987</div><div>2. Dikky Kurniawan - 152236035100-51</div><div>3. Elyadityo Santoso - 152236035101-567</div><div>4. Novri Kurniawan - 152236035100-1068</div><div>5. Robby Alfardo Irfan - 152236035100-380</div></div>

SELECTED THEME: Handwriting Recognition

TITLE OF THE PROJECT: Handwritten Name Recognition using CRNN

BACKGROUND SUMMARY:

Revolusi Industri 4.0 memudahkan industri untuk mendapatkan sumber data dan informasi. Berbagai cara dan metode dilakukan agar bisa mendapatkan insight yang berguna bagi bisnis maupun kehidupan sehari-hari. Salah satu metode yang terus berkembang adalah pengenalan teks pada gambar. Perusahaan-perusahaan besar seperti Google, Amazon, terus mengembangkan teknologi artificial intelligence untuk dapat dengan cepat mendeteksi gambar tulisan lalu mengekstrak tulisan tersebut. Fitur ini terus berkembang diantaranya diaplikasikan sebagai penunjuk jalan dan mengetahui kandungan makanan pada label kemasan. Namun, pengenalan teks pada tulisan tangan (handwritten) memiliki tantangan tersendiri karena beragam jenis tulisan mulai dari huruf bersambung atau tidak dan perbedaan karakteristik tulisan dari tiap orang. Oleh karena itu, topik ini menjadi hal yang menarik untuk dipelajari dan diteliti lebih jauh, sehingga proyek ini akan menjawab pertanyaan: “Bagaimana menangani pengenalan dari ribuan tulisan tangan dari jenis tulisan yang berbeda tiap orangnya melalui gambar?”

SCOPE:

Proyek ini akan dibatasi dalam pembuatan model untuk mengekstraksi fitur tulisan dari gambar berupa tulisan nama orang dari dataset yang telah disediakan. Hasil keluaran dari model akan dilakukan limitasi jumlah karakter. Oleh karena itu, pengaplikasian dari proyek ini dibatasi untuk rekognisi tulisan tangan dengan jumlah karakter terbatas.

PLEASE SPECIFICALLY MENTION WHAT YOU HAVE DONE:

1. Adi Kurniawan - 152236035100-987:

- Melakukan eksplorasi terhadap algoritma yang digunakan, CRNN + CTC
- Melakukan eksplorasi preparasi data untuk dapat digunakan
- Menyiapkan base model CRNN+CTC untuk dapat dituning/dioptimasi lebih lanjut

2. Dicky Kurniawan - 152236035100-51:

- Melakukan eksplorasi terhadap Single Evaluation Metrics yang akan digunakan
- Melakukan eksplorasi terhadap readability
- Melakukan tuning terhadap model dengan menggunakan overfitting/underfitting analysis pada performa model

3. Elyadityo Santoso - 152236035101-567

- Melakukan tuning terhadap base model CRNN yang sudah dibuat

4. Novri Kurniawan - 152236035100-1068:

- Melakukan deployment model menggunakan Streamlit dengan input gambar tulisan tangan

5. Robby Alfardo Irfan - 152236035100-380:

- Melakukan eksplorasi data
- Membuat powerpoint

HASIL Pengerjaan Proyek:

1. Business Understanding

a. Eksplorasi Literature Terkait Handwritten

Untuk melakukan pemahaman terhadap proyek secara keseluruhan dan mempelajari berbagai referensi terkait handwriting recognition. Masing-masing anggota membaca 2 paper atau dokumentasi di internet untuk melakukan pemahaman terhadap topik handwritten recognition.

b. Penentuan Single Evaluation Metrics

Selanjutnya, tim menentukan evaluation metric yang akan digunakan untuk mengevaluasi model. Berdasarkan Leung (2021), metric tradisional seperti akurasi, presisi,

recall, dan F1 kurang cocok dipakai untuk mengevaluasi text recognition. Berdasarkan berbagai literatur, Character Error Rate (CER) & Words Error Rate (WER) paling banyak digunakan untuk text recognition, dimana CER digunakan untuk menilai model dengan jumlah karakter sedikit, sedangkan WER digunakan untuk tulisan yang mengandung banyak kata2 (Panjang). Karena model ini memprediksi jumlah karakter yang rata-rata jumlahnya kurang dari 20 karakter, maka kami akan menggunakan CER sebagai single evaluation Metrics.

2. Data Cleaning

Pada proyek ini kami menggunakan dataset untuk training sebesar 100 ribu dan validasi sebesar 10 ribu. Pemilihan jumlah tersebut didasarkan pada variabilitas gambar dataset sudah terwakili dengan menggunakan jumlah tersebut, dan agar waktu training lebih efisien dari segi waktu dan power consumption. Selain itu, karakter juga kita batasi sebesar 2-16 mengingat jumlah karakter tersebut mewakili keseluruhan sampel pada dataset dan juga real world images yang coba akan kita prediksi.

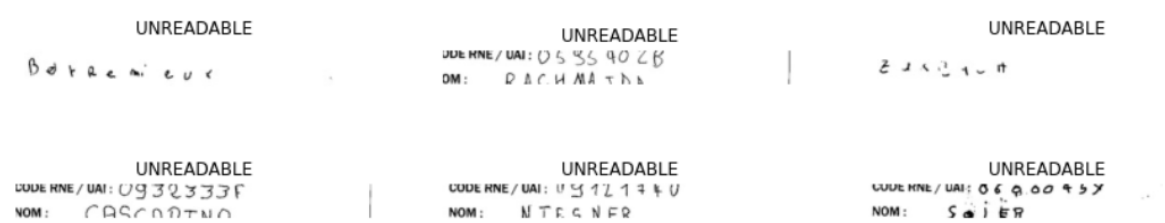
Beberapa cleaning yang kami lakukan terhadap dataset antara lain:

- a. Penanganan Missing Value: jumlah missing value yang ada pada dataset adalah 565 pada training, 78 pada validation, dan 70 pada testing. Karena jumlah dataset non NA masih cukup banyak dan variasi gambar dengan label missing masih dapat terwakili oleh sampel lainnya, maka kami memutuskan untuk menghapus label dengan nilai NA dari dataset.
- b. Penanganan Label dengan Huruf yang Tidak Kapital



Gambar tersebut menunjukkan bahwa gambar dengan label non kapital seharusnya memiliki label kapital. Oleh karena itu, kita ubah label non kapital menjadi kapital.

- c. Pananganan Label Unreadable



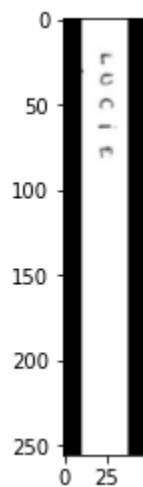
Berdasarkan eksplorasi yang kami lakukan, label unreadable pada gambar tidak konsisten. Mata manusia masih kesusahan membedakan mana gambar yang seharusnya dilabeli unreadable dan tidak. Kami telah mencoba menyusun model untuk memprediksi readability namun performanya kurang bagus. Oleh karena itu, kami mengeluarkan gambar dengan label unreadable dari training. Namun, kami tidak mengeluarkan dari testing karena kami ingin agar testing menunjukkan gambar adanya yang akan kita prediksi di real world.

3. Data Preparation

Dataset berupa kumpulan gambar terdapat dalam 3 folder berdasarkan jenisnya set training, set validasi, dan set tes. List dari gambar ini terangkum dalam file csv dilengkapi dengan label dari gambar tersebut sesuai dengan set kategori. Dari hasil eksplorasi terhadap dataset dilakukan cleaning terhadap sebagian kecil gambar dan label yang tidak sesuai dan berpotensi untuk merusak model yang akan di training. Set training memiliki lebih dari 300 ribu gambar lebih sementara test dan validation memiliki 40 ribu gambar lebih. Proses training menggunakan platform Kaggle.

a. Resize gambar

Untuk dapat diolah sebagai input dari model maka perlu dipastikan bahwa gambar memiliki dimensi yang sama. Dari hasil observasi terhadap dataset, memiliki variasi baik terhadap width maupun height dari gambar. Oleh karena itu perlu dilakukan resize terhadap gambar sesuai dengan ukuran yang ditentukan, misalnya 256x64 (WxH). Proses resize harus perlu diperhatikan agar gambar tulisan tidak mengalami distorsi sehingga sulit untuk dipelajari oleh model. Gambar yang sudah di resize perlu untuk dilakukan rotasi 90 derajat searah jarum jam sehingga akan lebih mudah untuk dilakukan ekstraksi terhadap fitur karakternya.



b. Padding Label

Dari hasil pengecekan terhadap karakter yang digunakan pada label yang telah di ubah ke huruf kapital. Didapati 29 kelas karakter yang akan menjadi referensi sebagai kelas output CNN: [' ', "'", '-', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K', 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X', 'Y', 'Z']. Masing-masing karakter akan dikodifikasi dalam bentuk integer 1-29.

Label yang akan masuk ke modul RNN memiliki shape yang sama. Kita dapat set bahwa input label maksimal yang digunakan adalah 16 karakter. Label yang memiliki lebih dari 16 karakter tidak digunakan, sementara label yang memiliki kurang dari 16 karakter akan dipadding dengan suatu integer token di luar dari kelas karakter yang ada, kita set 99.

c. Data Pipeline

Data gambar training membutuhkan resource yang besar. Oleh karena itu proses training dapat dioptimalisasi dengan menggunakan pipeline tensorflow `tf.data.Dataset`. Data input dan label harus diubah menjadi struktur data tensor. Struktur data ini didukung dengan library yang cukup lengkap sehingga dapat memudahkan untuk melakukan manipulasi data gambar berupa membersihkan data, scaling data, atau transformasi data.

4. Model Building

Model yang dipakai untuk melakukan handwritten recognition adalah CRNN dengan perhitungan loss function menggunakan CTC loss.

$$\text{Model} = \text{CNN} + \text{RNN} + \text{CTC loss}$$

Model terdiri dari tiga bagian penting:

1. Convolution Neural Network (CNN) untuk mengekstrak fitur pada gambar
2. Recurrent Neural Network (RNN) untuk melakukan prediksi urutan output tiap satuan waktu
3. Connectionist Temporal Classification Loss (CTC loss) adalah tambahan layer yang digunakan untuk memprediksi probabilitas output setiap satuan waktu

a. Convolution Neural Network (CNN)

Pada basic model terdapat dilakukan beberapa operasi berupa konvolusi, max pooling, dan Dense. Input dari modul ini berupa gambar dengan ukuran 256 x 64 pixel. Jika pada algoritma CNN biasa akan menghasilkan output berupa kelas dari klasifikasi yang diinginkan seperti apakah gambar kucing atau anjing, maka pada kasus yang membutuhkan sequence seperti halnya OCR atau handwriting recognition ini maka akan diolah terlebih dahulu oleh RNN untuk menentukan state atau sequential dari fitur-fitur karakter yang terdeteksi.

b. Recurrent Neural Network (RNN)

RNN yang digunakan pada contoh ini adalah Bidirectional LSTM. Matrix Vector data hasil proses konvolusi dan max pooling pada CNN lalu diteruskan sebagai input pada RNN. Vector ini lalu dicocokkan dengan target kelas karakter yang telah ditentukan sehingga menjadi sequence karakter atau kata. Untuk menentukan seberapa akurat data label dan data prediksi di layer berikutnya ditambahkan suatu transkriptif layer untuk menghitung loss function.

c. Connectionist Temporal Classification (CTC)

Dalam proses pengenalan kata atau kalimat dalam sebuah gambar perlu untuk mengetahui 2 hal, lokasi dari karakter pembentuk kata dan juga encoding dari kata tersebut. Bentuk tulisan setiap orang sangat bervariasi. Akan mengalami kendala untuk mengetahui bahwa karakter ini adalah urutan huruf ketiga atau keempat dari kata yang dicari. Apakah terjadi pengulangan dalam karakter tersebut.

Secara sederhana CTC membantu untuk menyelesaikan permasalahan tersebut. Algoritma CTC terbagi dalam 2 tahapan [3]:

1. Train:

Pada tahapan ini kita akan menambahkan satu transcriptive layer untuk menghitung *loss function*. Dari sequence yang diprediksi oleh RNN hanya berupa matrik nilai dari karakter dari tiap sequence element.

2. Infer

Vector hasil prediksi dari CTC ini nantinya perlu untuk dilakukan decoding. Ada beberapa CTC jenis decoding yang biasa digunakan. Dalam tugas ini kita menggunakan *greedy algorithm* atau biasa disebut *best path decoding*. Algoritma ini adalah algoritma decoding paling sederhana dan cukup untuk digunakan pada tugas yang tidak begitu complex.

CTC mengenalkan *pseudo character* yang dilambangkan dengan ‘-’ untuk mengatasi permasalahan terhadap duplikasi karakter. Hal ini disebabkan proses ekstraksi dari fitur CNN tidak memberikan lokasi dari karakter tersebut dalam kata, atau yang biasa disebut *alignment*. Oleh karena itu perlu ditambahkan 1 atau 2 kelas karakter tambahan dari jumlah vocabulary characters yang digunakan. Sebagai contoh untuk membedakan “to” dan “too”

- “to” → “---ttttttooo”, “-t-o-”, “to”
- “too” → “---ttttto-o”, “-t-o-o-”, “to-o”, tetapi **bukan** “too”

CTC juga banyak digunakan pada aplikasi aplikasi yang membutuhkan *alignment* seperti voice recognition.

Base model yang digunakan pada proyek ini mengadopsi dari contoh yang disediakan pada website keras [1], [2].

Model: "handwriting_recognizion"

Layer (type)	Output Shape	Param #	Connected to
image (InputLayer)	[(None, 64, 256, 1)]	0	
Conv1 (Conv2D)	(None, 64, 256, 32)	320	image[0][0]
pool1 (MaxPooling2D)	(None, 32, 128, 32)	0	Conv1[0][0]
Conv2 (Conv2D)	(None, 32, 128, 64)	18496	pool1[0][0]
pool2 (MaxPooling2D)	(None, 16, 64, 64)	0	Conv2[0][0]
reshape (Reshape)	(None, 64, 1024)	0	pool2[0][0]
dense1 (Dense)	(None, 64, 64)	65600	reshape[0][0]
dropout_2 (Dropout)	(None, 64, 64)	0	dense1[0][0]
bidirectional_4 (Bidirectional)	(None, 64, 256)	197632	dropout_2[0][0]
bidirectional_5 (Bidirectional)	(None, 64, 128)	164352	bidirectional_4[0][0]
label (InputLayer)	[(None, None)]	0	
dense2 (Dense)	(None, 64, 31)	3999	bidirectional_5[0][0]
ctc_loss (CTCLayer)	(None, 64, 31)	0	label[0][0] dense2[0][0]
Total params: 450,399			
Trainable params: 450,399			
Non-trainable params: 0			

5. Model Tuning & Model Evaluation

a. Penentuan Bias Optimal Error

Sebelum melakukan analisis terhadap model, kami menentukan bias optimal error terlebih dahulu. Kami melakukan pengecekan terhadap 300 gambar, dan didapat bahwa rata-rata terdapat 3 gambar yang susah kami mengerti. Oleh karena itu, kita menentukan bias optimal error sebanyak 1%.

b. Model Tuning & Evaluation

Kami menggunakan teknik Orthogonalization untuk melakukan tuning pada model. Model I memiliki CER pada training sebesar 6.1% dan CER pada validation sebesar 6.8%. Selisih error yang besar pada training dan bias optimal error menandakan bahwa model masih mengalami underfitting. Oleh karena itu, kami akan fokus membenahi masalah underfitting terlebih dahulu.

Karena model I masih memiliki masalah underfitting, kami akan mencoba mengatasi masalah underfitting tersebut pada model II dengan menambah CNN layer menjadi 4 layer, dan dense layer menjadi 2 layer. Hasilnya, Model II mampu meningkatkan CER pada training dataset menjadi sebesar 2.93%. Namun, performa pada validation dataset CERnya masih di angka 5.67%. Hal ini mengindikasikan bahwa model masih memiliki sedikit masalah overfitting. Oleh karena itu, pada model selanjutnya kita akan berfokus menangani masalah underfitting tersebut.

Untuk mengatasi masalah overfitting pada model II, kami mencoba mengatasinya dengan menambahkan layer drop out. Cara tersebut cukup berhasil dimana Model III memiliki CER sebanyak 3% untuk training set dan sekitar 4% untuk validation dan testing set. Mengingat OCR ini melibatkan karakter tulisan tangan yang kadang susah dikenali bahkan oleh manusia, maka kami merasa error rate tersebut dapat diterima dan akan dibandingkan dengan model lainnya.

c. Error Analysis

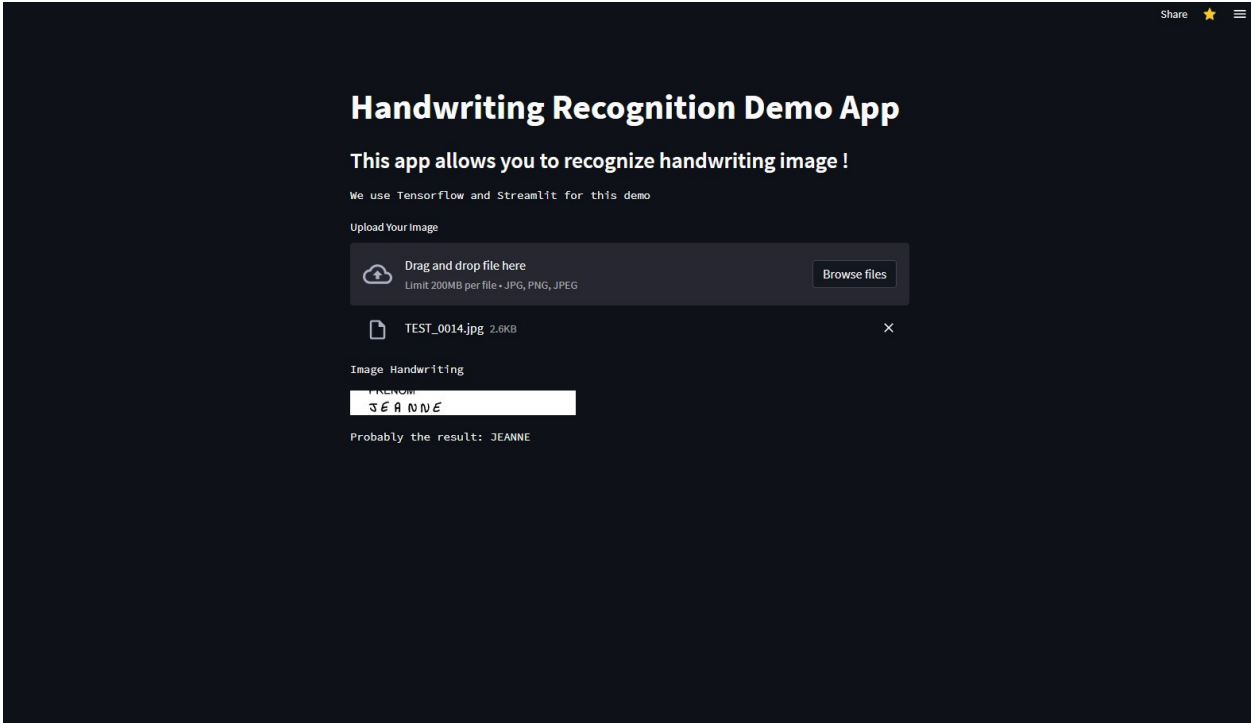
Di sini, kami melakukan perangkingan terhadap gambar yang memiliki CER tertinggi (tingkat kesalahan tertinggi) dan melakukan analisis mengapa hal tersebut bisa terjadi. Berikut gambar-gambar yang memiliki nilai CER tertinggi.



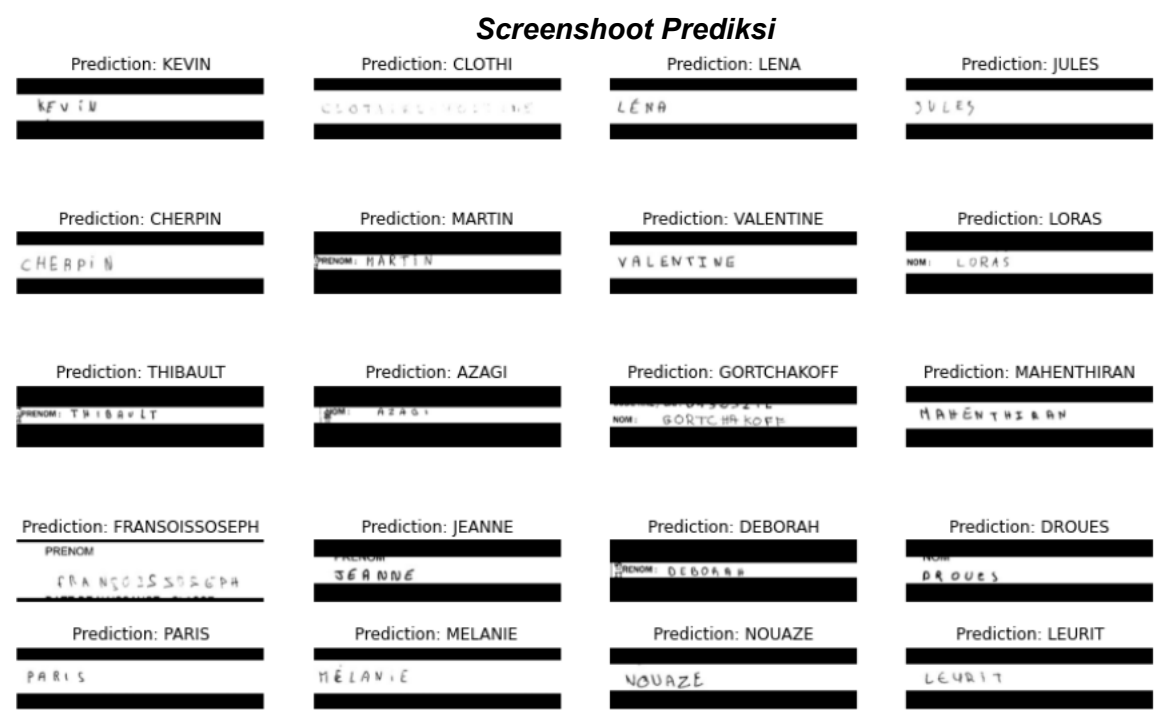
Terlihat bahwa kebanyakan gambar yang salah terprediksi dengan nilai CER besar karena memang tulisan tersebut sulit dibaca oleh mata manusia atau karena gambar tersebut terjadi kesalahan dalam pelabelan.

6. Deployment

Setelah mendapatkan model yang memiliki performa maksimal, model kami aplikasikan ke dalam aplikasi web yang dapat memprediksi tulisan tangan. Adapun deployment model dilakukan dengan menggunakan Streamlit. Pengguna dapat mengupload gambar tulisan dan memprediksi hasil rekognisi nya. Selain itu, kami juga menyediakan canvas bagi pengguna yang ingin menulis langsung dan memprediksi hasil rekognisi dari tulisannya.



SCREENSHOOT OUTPUT MODEL:
Foto/screenshot yang menunjukkan bukti model bekerja.



Screenshoot Performa Model pada Test Dataset

	FILENAME	IDENTITY	PREDICTION	CER	WER
0	TEST_0001.jpg	KEVIN	KEVIN	0.0	0.0
1	TEST_0002.jpg	CLOTAIRE	CLOTHI	37.5	100.0
2	TEST_0003.jpg	LENA	LENA	0.0	0.0
3	TEST_0004.jpg	JULES	JULES	0.0	0.0
4	TEST_0005.jpg	CHERPIN	CHERPIN	0.0	0.0
...
3195	TEST_3203.jpg	ENZO	ENZO	0.0	0.0
3196	TEST_3204.jpg	VALENTIN	VALENTIN	0.0	0.0
3197	TEST_3205.jpg	DUMONT	DUMONT	0.0	0.0
3198	TEST_3206.jpg	JUSTINE	JUSTINE	0.0	0.0
3199	TEST_3207.jpg	BROUQUIER	BROUQUIER	0.0	0.0

3200 rows × 5 columns

CER_test_average:

4.27765584375

WER_test_average:

15.77604165625

GITHUB REPO LINK:
<https://github.com/kurniawan2805/handwriting-recognition>

10-Min VIDEO PRESENTATION LINK:
<https://youtu.be/ZUAQaOMqx8M>

PROJECT REFERENCES:
Dataset apa saja yang digunakan, makalah/jurnal/artikel, dan sumber daya lain yang diperlukan untuk memenuhi tujuan proyek.
Dataset : Data Handwriting
(<https://drive.google.com/drive/folders/1izbmpOCkNmVudpUVOMiVV7zW6Pt511p1?usp=sharing>)

Jurnal :

- XinSheng, Z., Yu, W. 2022. *Industrial character recognition based on improved CRNN in complex environments*. Computers in Industry 142. <https://doi.org/10.1016/j.compind.2022.103732>
- Hao Zeng, 2020. *An Off-line Handwriting Recognition Employing Tensorflow*. <https://doi.org/10.1109/ICBAIE49996.2020.00040>
- Weibo Yu; Chunyu Guo; Keping Liu; Hongtao Yang. 2020. *Handwritten Digital Recognition Optimization Method based on Deep Learning*. <https://doi.org/10.1109/CAC51589.2020.9327647>
- SaraAqab, Muhammad Usman Tariq. 2020. *Handwriting Recognition using Artificial Intelligence Neural Network and Image Processing*. <https://doi.org/10.14569/IJACSA.2020.0110719>
- G.R. Hemanth, M. Jayasree, S. Keerthi Venii, P. Akshaya, and R. Saranya. 2021. *CNN-RNN Based Handwritten Text Recognition*. <https://doi.org/10.21917/ijsc.2021.0351>

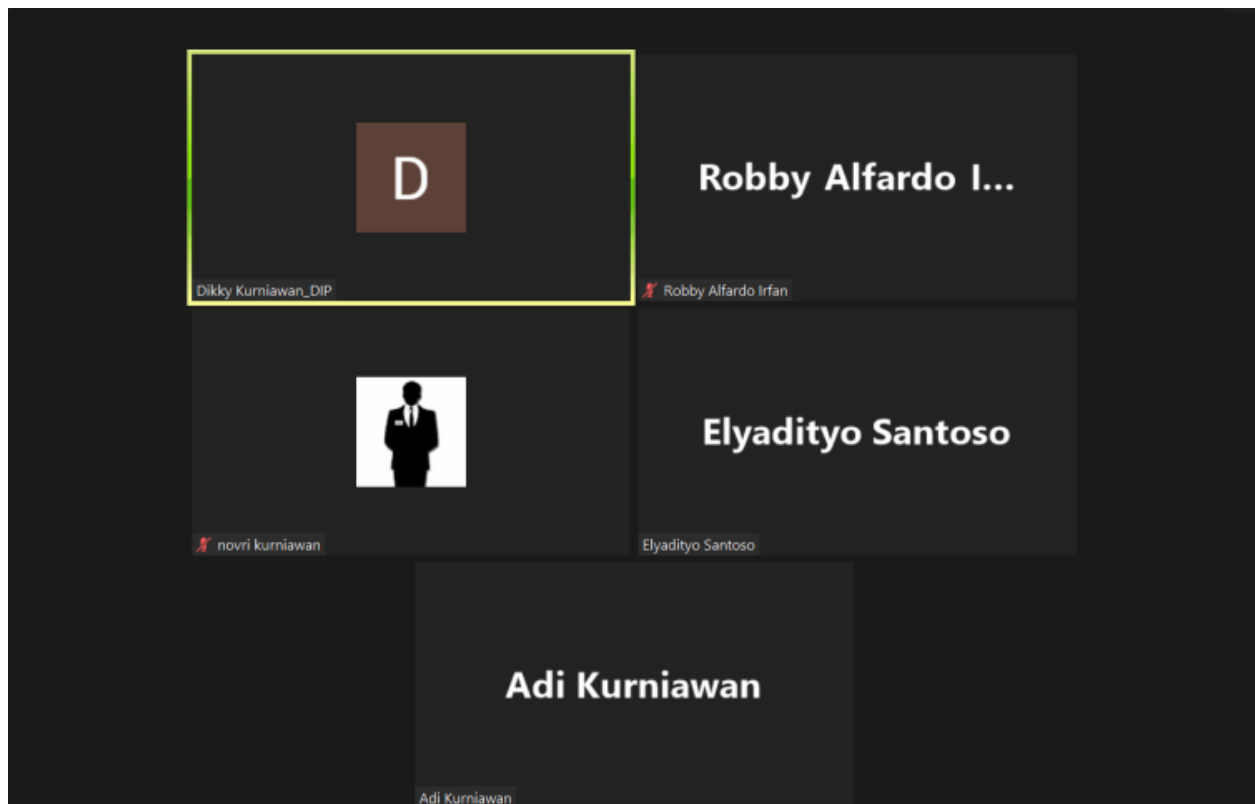
Artikel:

1. https://keras.io/examples/vision/captcha_ocr/
2. https://keras.io/examples/vision/handwriting_recognition/
3. <https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c>
4. <https://towardsdatascience.com/evaluating-ocr-output-quality-with-character-error-rate-cer-and-word-error-rate-wer-853175297510>

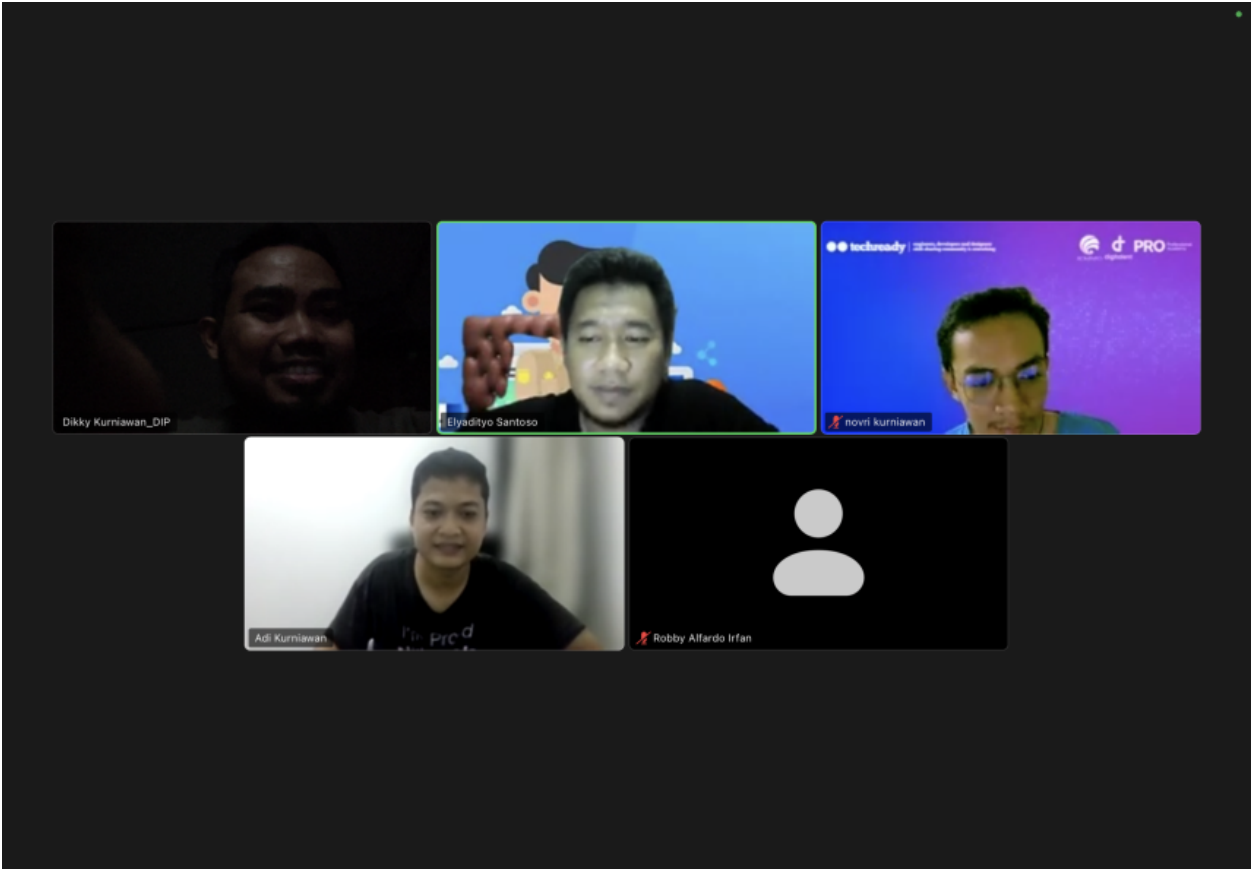
DOCUMENTATION:

Foto/screenshot/bukti lain yang menunjukkan proses pengerjaan project Demo Day selama waktu yang diberikan. Screenshot boleh berupa foto zoom bersama, atau group chat. Minimal 3 bukti/foto.

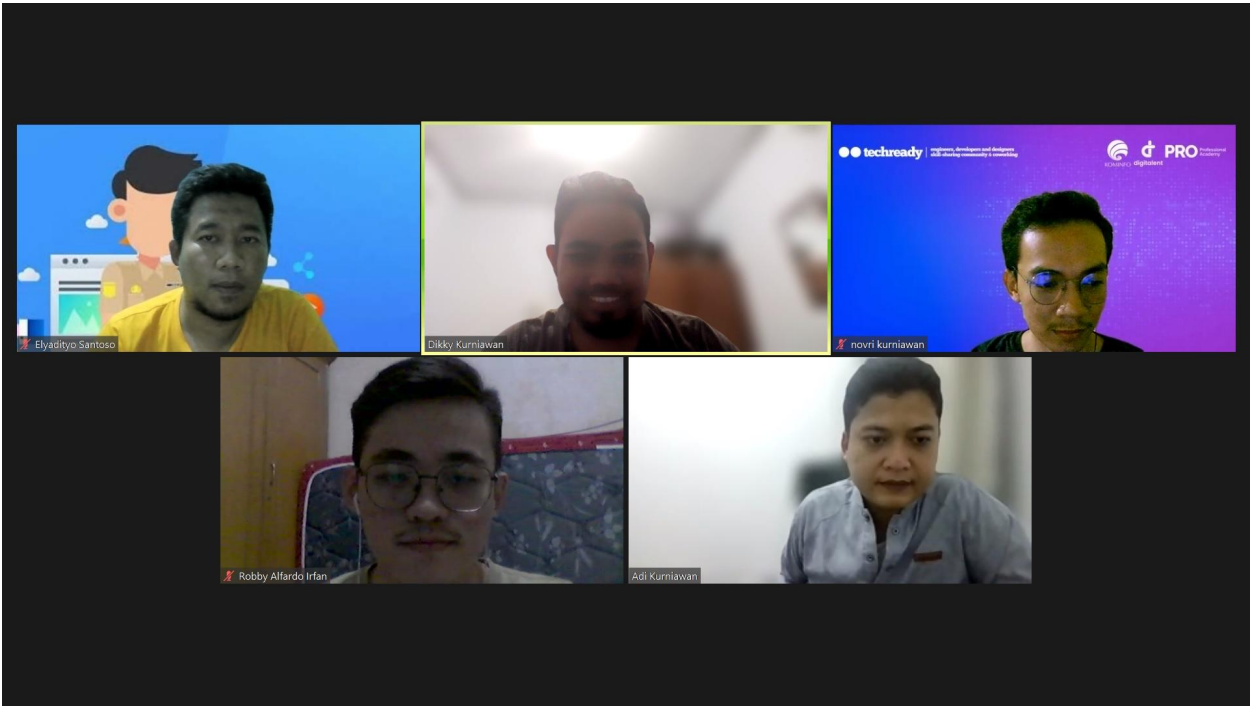
Pertemuan Pertama: Pembagian Tugas & Eksplorasi Dataset



Pertemuan Kedua: Eksplorasi Model




Pertemuan Ketiga: Model Evaluation, Pembuatan Laporan, PPT, Video



Screenshot diskusi dan notulen meeting di Discord:

welcome-and-rules


yukk



Dikky Kurniawan

07/16/2022

Yuk, udah tak buka zoomnya



robby


07/16/2022

Meeting 16 Juli 2022
Pukul 09.00 WIB
Hadir : Adi, Dikky, Novri, Robby

Notes:

- Seragamin Cleaning
(Take out Readable, NA, desk-desk)
- Train data yg benar-benar bersih
- Video digabung dari (Perkenalan, job masing-masing, penjelasan dataset, hasil presentasi proyek dan komparasi, penutup) dalam waktu 10 menit
- Meeting Berikutnya Sabtu, 23 Juli 2022 pukul 19.30 WIB
- Contoh code mas adi bisa 1 jam dalam run modelnya (Jadi estimasi waktu dan hindarin plagiarism)

Semangat rekan-rekan!!!




Adi Kurniawan

07/16/2022


<https://www.kaggle.com/code/accountstatus/handwriting-recognition>

Handwriting Recognition

Explore and run machine learning code with Kaggle Notebooks | Using data from Handwriting Recognition



ini ref yg tadi



robby

07/23/2022

Meeting 23 Juli 2022
Pukul 19.30 - Selesai
Hadir: All


Topik: Pembahasan Modelling

Notes:

- Perlu Resize gambar biar sama train dan test
- Eksplor crr & wrr
- Eksplor Transfer Learning
- Komparasi model
- Membuat Power Point

Website Source:
https://keras.io/examples/vision/captcha_ocr/
<https://github.com/githubharald/SimpleHTR>
<https://towardsdatascience.com/build-a-handwritten-text-recognition-system-using-tensorflow-2326a3487cd5>
<https://github.com/tienthienhd/tf-crnn-ctc/blob/master/model.py>
<https://www.kaggle.com/code/kurniawan2805/ohr-pipeline>

Keras documentation: OCR model for reading Captchas

 Keras

GitHub

GitHub - githubharald/SimpleHTR: Handwritten Text Recognition (HTR)...

Handwritten Text Recognition (HTR) system implemented with TensorFlow. - GitHub - githubharald/SimpleHTR: Handwritten Text Recognition (HTR) system implemented with TensorFlow.