



UNDERGRADUATE THESIS

**DETECTION OF THE INDONESIAN SIGN
LANGUAGE (BISINDO) ALPHABET USING THE
DETECTION TRANSFORMER (DETR)**

MUKHAMAD AZIZ FIRMANSYAH
NPM 22081010030

THESIS ADVISORS

Fetty Tri Anggraeny, S.Kom., M.Kom.
Yisti Vita Via, S.ST., M.Kom.

**MINISTRY OF HIGHER EDUCATION, SCIENCE, AND TECHNOLOGY
UNIVERSITAS PEMBANGUNAN NASIONAL VETERAN JAWA TIMUR
FACULTY OF COMPUTER SCIENCE
INFORMATICS STUDY PROGRAM
SURABAYA
2026**

APPROVAL SHEET

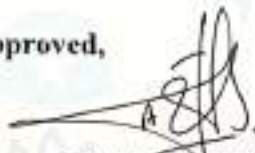
DETECTION OF THE INDONESIAN SIGN LANGUAGE (BISINDO) ALPHABET USING THE DETECTION TRANSFORMER (DETR)

By:
MUKHAMAD AZIZ FIRMANSYAH
NPM. 22081010030

Has been defended before, and accepted by, the Board of Assessors of the Thesis Examination of the Informatics Study Program, Faculty of Computer Science, Universitas Pembangunan Nasional Veteran Jawa Timur, on May 7, 2026:

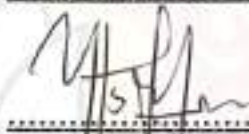
Approved,

Fetty Tri Anggraeny, S.Kom., M.Kom.
NIP. 19820211 202121 2 005



(Advisor I)

Yisti Vita Via, S.ST., M.Kom.
NIP. 19860425 202121 2 001



(Advisor II)

Eka Prakarsa Mandvartha, S.T., M.Kom.
NIP. 19880525 201803 1 001



(Head Assessor)

Chrystia Aji Putra, S.Kom., M.T.
NIP. 19861008 202121 1 001



(Assessor I)

Acknowledge by,

Dean of the Faculty of Computer Science



Prof. Dr. Ir. Novirina Hendrasarie, M.T.
NIP. 19681126 199403 2 001

APPROVAL SHEET

**DETECTION OF THE INDONESIAN SIGN LANGUAGE (BISINDO)
ALPHABET USING THE DETECTION TRANSFORMER (DETR)**

By:
MUKHAMAD AZIZ FIRMANSYAH
NPM. 22081010030

Approved to proceed to the Thesis Examination

Approved by,

Coordinator of Informatics Study Program
Faculty of Computer Science



Dr. Intan Yuniar Purbasari, S.Kom., M.Sc.
NIP. 19800602 202521 2 029

STATEMENT OF ORIGINALITY

I am the undersigned:

Student Name : Mukhamad Aziz Firmansyah
NPM : 22081010030
Degree Program : Bachelor (S1)
Study Program : Informatics
Faculty : Faculty of Computer Science

Hereby declares that this undergraduate thesis contains no part of any other scientific work that has been submitted to obtain an academic degree at any higher education institution. Furthermore, it does not contain any work or opinions previously written or published by others, except for those which are explicitly cited in this thesis and listed completely in references.

And I declare that this scientific document is free from elements of plagiarism. If in the future indications of plagiarism are found in this Thesis, I am willing to accept sanctions in accordance with the applicable laws and regulations.

Thus, I made this statement without any coercion from anyone and to be used as it should.



Surabaya, May 21, 2026
Declarant,



MUKHAMAD AZIZ FIRMANSYAH
NPM. 21081010086

ABSTRACT

Student Name / NPM: Mukhamad Aziz Firmansyah/22081010030
Thesis Title: DETECTION OF THE INDONESIAN SIGN LANGUAGE (BISINDO) ALPHABET USING THE DETECTION TRANSFORMER (DETR)
Supervisor: 1. Fetty Tri Anggraeny, S.Kom., M.Kom.
2. Yisti Vita Via, S.ST., M.Kom.

Indonesian Sign Language (BISINDO) is the primary language used by the deaf community in Indonesia. However, the communication gap between BISINDO users and the general public remains a challenge that has not yet been fully resolved. This study develops a real-time BISINDO alphabet detection system using the Detection Transformer (DETR) architecture with a ResNet-50 backbone, featuring a single-layer transformer encoder-decoder and 25 object queries, trained to recognize 26 letters (A–Z) and 17 BISINDO words. The research was conducted under two training scenarios: Scenario 1 without on-the-fly augmentation and Scenario 2 with on-the-fly augmentation, which includes spatial and photometric transformations. Evaluation was performed using standard PyCOCOTools metrics, along with real-time robustness testing under three conditions: normal conditions with the same subject, dynamic lighting variations, and subjects different from the training data. PyCOCOTools evaluation results show that Scenario 2 with augmentation outperforms in nearly all key metrics: AP@[IoU=0.50:0.95] of 0.7188 (vs 0.6772), AP@0.75 of 0.9192 (vs 0.8118), AR@maxDets=100 of 0.7949 (vs 0.7219), with the most significant improvements observed for medium-sized objects, where the medium AP increased by 27.6% and the medium AR by 31.7%. In real-time testing, Scenario 2 maintained a 100% detection success rate across all test conditions, while Scenario 1 only achieved 34.9% under the most challenging condition, different subjects with varying lighting. The developed system runs at 30–60 FPS on a GPU and 5–9 FPS on a CPU, and is implemented in a web-based interface using Streamlit. This study concludes that on-the-fly augmentation plays a crucial role in building models that are not only superior in metrics but also reliable and generalize well to real-world conditions.

Keywords: Alphabet, BISINDO, DETR, On-the-Fly Augmentation, Real Time, Streamlit

ACKNOWLEDGEMENTS

The author offers heartfelt thanks and praise to Allah SWT for all His mercy and blessings, which have enabled the author to complete this thesis titled **“DETECTION OF THE INDONESIAN SIGN LANGUAGE (BISINDO) ALPHABET USING THE DETECTION TRANSFORMER (DETR)”**. Throughout the process of writing this thesis, the author realized that many people have provided assistance, support, and motivation, both directly and indirectly. Therefore, with the utmost respect, the author would like to express gratitude to:

1. Prof. Dr. Ir. Akhmad Fauzi, M.M.T., IPU., Rector of Universitas Pembangunan Nasional “Veteran” Jawa Timur.
2. Prof. Dr. Ir. Novirina Hendrasarie, M.T., Dean of the Faculty of Computer Science, Universitas Pembangunan Nasional “Veteran” Jawa Timur..
3. Dr. Intan Yuniar Purbasari, S.Kom., M.Sc., Coordinator of the Informatics Study Program, Universitas Pembangunan Nasional “Veteran” Jawa Timur.
4. Budi Nugroho, S.Kom., M.Kom., as the Academic Advisor of the author, for his guidance, direction, and motivation throughout the period of study.
5. Fetty Tri Anggraeny, S.Kom., M.Kom., and Yisti Vita Via, S.ST., M.Kom., as Thesis Advisors, for their guidance, direction, motivation, and valuable advice throughout the research and writing process.
6. Eka Prakarsa Mandyartha, S.T., M.Kom., and Chrystia Aji Putra, S.Kom., M.T., as Examiners, for providing meaningful feedback, suggestions, and evaluations to improve this thesis.
7. All lecturers of Universitas Pembangunan Nasional “Veteran” Jawa Timur, especially those in the Informatics Study Program, as well as the faculty staff and administrative personnel, for their knowledge, guidance, and valuable support throughout my academic journey.
8. My parents and family, who have always provided encouragement, unceasing prayers, and endless love.
9. Meidy Alesya, who assisted with testing this research application and provided encouragement and motivation throughout the thesis writing process.

10. My classmates from the 2022 Computer Science Program cohort, for their support and cooperation from the start of our studies through to the completion of this thesis.
11. Warkop Alang and Mamino, which served as places for discussion and working on assignments during college, together with friends (Nobita Group, Alang Technology, and Living Legends Jomokerto).
12. All parties who cannot be mentioned individually, for their prayers, assistance, and support.

The author acknowledges that there are still shortcomings and limitations in this research. Therefore, the author sincerely welcomes all constructive criticism, suggestions, and feedback to improve this thesis report.

Surabaya, May 21, 2026



Mukhamad Aziz Firmansyah
NPM. 22081010030

TABLE OF CONTENTS

COVER PAGE	iii
APPROVAL SHEET	v
APPROVAL SHEET	vii
STATEMENT OF ORIGINALITY	ix
ABSTRACT	xi
ACKNOWLEDGEMENTS	xiii
TABLE OF CONTENTS	xv
LIST OF FIGURES	xix
LIST OF TABLES	xxiii
LIST OF CODE PROGRAMS	xxv
LIST OF APPENDICES	xxvii
LIST OF NOTATIONS	xxix
CHAPTER I INTRODUCTION	1
1.1 Background	1
1.2 Research Question	3
1.3 Research Objectives	3
1.4 Research Benefits	4
1.5 Scope of the Study	4
CHAPTER II LITERATURE REVIEW	7
2.1 Previous Research	7
2.2 Indonesian Sign Language (BISINDO)	11
2.3 Machine Learning	12
2.4 Deep Learning	13
2.5 Computer Vision	14
2.6 Object Detection	15
2.7 Detection Transformer (DETR)	16
2.7.1 DETR Basic Concept	16
2.7.2 DETR Architecture	17
2.7.3 Bipartite Matching with Hungarian Algorithm	21

2.8 On-the-Fly Data Augmentation	25
2.9 Real-Time Processing	26
2.10 Libraries and Frameworks.....	26
2.10.1 PyTorch.....	27
2.10.2 OpenCV	27
2.10.3 Albumentations	27
2.10.5 Label Studio	28
2.10.6 Streamlit	28
2.11 Evaluation	29
2.11.1 Object Detection Model Evaluation.....	29
2.11.2 Computational Evaluation.....	30
CHAPTER III RESEARCH METHODOLOGY	33
3.1 System Design.....	33
3.2 Tools and Materials.....	34
3.2.1 Hardware Requirements.....	34
3.2.2 Software Requirements	34
3.2.3 Dataset Sources	34
3.3 Data Collection.....	35
3.3.1 Data Collection Procedure	35
3.3.2 Data Validation	37
3.4 Data Annotation	37
3.5 Data Preprocessing.....	42
3.5.1 Purpose and Workflow of Data Preprocessing	42
3.5.2 Data Preprocessing Steps	42
3.6 Data Augmentation	45
3.6.1 On-the-Fly Data Augmentation	45
3.6.2 Difference Between Training vs Testing/Validation Pipeline	49
3.6.3 Safe Transform Mechanism and Error Handling	51
3.7 DETR	53
3.7.1 Image Input and Feature Extraction by Backbone.....	55
3.7.2 Dimension Projection (1x1 Conv) and Flattening	57
3.7.3 Spatial Positional Encoding	58

3.7.4 Transformer Encoder	59
3.7.5 Object Queries and Transformer Decoder	61
3.6.7 Prediction Heads	69
3.6.8 Bipartite Matching and Loss Function.....	70
3.8 Evaluation	78
3.8.1 Object Detection Metric Evaluation.....	78
3.8.2 Real-Time Implementation Evaluation.....	78
3.9 Implementation	79
3.9.1 Web-Based System Implementation.....	79
3.9.2 Real-Time Detection Implementation.....	82
3.9 Experiment Scenarios.....	86
CHAPTER IV RESULTS AND DISCUSSION	89
4.1 Data Collection and Preparation Results.....	89
4.1.1 Data Acquisition Results.....	89
4.1.2 Characteristics of the Generated Dataset	92
4.1.3 Data Annotation Results	92
4.2 Data Preprocessing Results	95
4.3 On-the-Fly Data Augmentation Results.....	97
4.4 DETR Model Training	99
4.4.1 Training Configuration	99
4.4.2 Model Training Process	100
4.4.3 Model Training Results Scenario 1 (Without Augmentation)	101
4.4.4 Model Training Results Scenario 2 (With On-the-Fly Augmentation)	106
4.5 Experiment Scenario I Results (Without Augmentation)	111
4.5.1 PyCOCOTools Evaluation Scenario 1 (Without Augmentation)	111
4.5.2 Model Test Data Detection Results Scenario 1 (Without Augmentation)	116
4.6 Experiment Scenario II Results (With On-the-Fly Augmentation)	118
4.6.1 PyCOCOTools Evaluation Scenario 2 (With On-the-Fly Augmentation)	118
4.6.2 Model Test Data Detection Results Scenario 2 With Augmentation.	124

4.7 Comparison and Analysis Between Scenarios.....	126
4.7.1 Comparison of Training Characteristics Between Scenarios.....	126
4.7.2 Comparison of PyCOCOTools Evaluation Results Between Scenarios	127
4.8 Web-Based System Implementation.....	128
4.8.1 Web System Architecture and Features.....	129
4.8.2 Integration of Model and Inference Pipeline.....	130
4.8.3 Deployment on Streamlit Community Cloud.....	132
4.8.4 Web System Feasibility Analysis.....	132
4.9 Real-Time System Implementation and Evaluation.....	133
4.9.1 Real-Time Detection Implementation.....	133
4.9.2 Computational Performance Evaluation.....	135
4.9.3 Detection Stability Evaluation Under Normal Conditions.....	136
4.9.4 Lighting Robustness Evaluation.....	141
4.9.5 Subject Variation Robustness Evaluation.....	148
4.10 Summary of Robustness Evaluation.....	156
CHAPTER V CONCLUSION AND RECOMMENDATION.....	161
5.1 Conclusion.....	161
5.2 Recommendation.....	162
REFERENCES.....	165
Appendix 1. Dataset.....	175
Appendix 2. Annotation Visualization.....	181
Appendix 3. Detection Results on Test Data (Scenario 1).....	185
Appendix 4. Detection Results on Test Data (Scenario 2).....	189
Appendix 5. Visualization of Real-Time Detection Results Comparison Under Normal Conditions.....	193
Appendix 6. Visualization of Real-Time Detection Results Comparison Under Different Lighting Conditions.....	203
Appendix 7. Different Subject Robustness Testing.....	213

LIST OF FIGURES

Figure 2.1 Alfabet BISINDO [19]	11
Figure 2.2 DETR Basic Concept [12]	17
Figure 2.3 DETR Architecture [12]	18
Figure 2.4 Transformer Architecture in DETR [12]	20
Figure 3.1 System Design	33
Figure 3.2 Creating a Project Folder in Label Studio	37
Figure 3.3 Data Annotation Workflow	38
Figure 3.4 Labeling Setup	39
Figure 3.5 Drawing the Bounding Box for the Alphabet A.....	40
Figure 3.6 Images Folder After Extraction	41
Figure 3.7 Labels Folder After Extraction	41
Figure 3.8 Data Preprocessing Workflow	42
Figure 3.9 On-the-Fly Data Augmentation Workflow.....	45
Figure 3.10 DETR Architecture (adapted from [66])	54
Figure 3.11 Backbone DETR.....	55
Figure 3.12 Image Input.....	55
Figure 3.13 Visualization of Channels with Highest Variance and Mean Activation in Each Layer of ResNet-50.....	56
Figure 3.14 Visualization of Dimension Projection and Feature Map Flattening	57
Figure 3.15 Transformer DETR.....	58
Figure 3.16 2D Visualization of Sinusoidal Positional Encoding	59
Figure 3.17 Visualization of the Transformer Encoder Mechanism on 49 Feature Tokens	60
Figure 3.18 Representation of Learned Object Queries in the Transformer Decoder	61
Figure 3.19 Visualization of Self-Attention Among Object Queries in the Transformer Decoder	63
Figure 3.20 Distribution of Query Cross-Attention on the Encoder Feature Map	65
Figure 3.21 Final Embedding Representations for Object Queries in the Decoder	68
Figure 3.22 Prediction Heads and Bipartite Matching.....	68

Figure 3.23 25×44 Logits Heatmap, Bar Chart of Probability per Query, 25 Bboxes before Filtering.....	69
Figure 3.24 Detection Results (Query 18)	76
Figure 3.25 Streamlit Website Wireframe	79
Figure 3.26 Website Implementation Workflow	80
Figure 3.27 Real-Time Implementation Workflow	83
Figure 4.1 BISINDO Alphabet A Dataset Acquisition Process.....	91
Figure 4.2 BISINDO Alphabet A Data Acquisition Results.....	91
Figure 4.3 Annotation Process for Alphabet A Using Label Studio.....	93
Figure 4.4 Training and Validation Loss Curves for Scenario 1	101
Figure 4.5 Log Loss for Scenario 1: Early Training Epoch.....	102
Figure 4.6 Log Loss for Scenario 1: Mid-Training Epochs	103
Figure 4.7 Log Loss for Scenario 1: Continued Training Epochs	104
Figure 4.8 Log Loss for Scenario 1: Final Training Epochs.....	105
Figure 4.9 Training and Validation Loss Curve for Scenario 2.....	106
Figure 4.10 Log Loss for Scenario 2: Early Training Epochs	107
Figure 4.11 Log Loss for Scenario 2: Mid-Training Epochs.....	108
Figure 4.12 Log Loss for Scenario 2: Continued Training Epochs	109
Figure 4.13 Log Loss for Scenario 2: Final Training Epochs.....	110
Figure 4.14 Evaluation of the PyCOCOTools Model for Scenario 1	113
Figure 4.15 Detection Results Alphabet A (Scenario 1 Without Augmentation).....	117
Figure 4.16 Evaluation of the PyCOCOTools Model for Scenario 2	120
Figure 4.17 Detection Results Alphabet A (Scenario 2, With Augmentation).....	124
Figure 4.18 Streamlit Application Interface on Desktop View.....	129
Figure 4.19 Streamlit Application Interface on Mobile View	130
Figure 4.20 Detection Results in the Streamlit Application	131
Figure 4.21 Deployment Process on Streamlit Community Cloud.....	132
Figure 4.22 Example of Model Detection Failure in Scenario 1	137
Figure 4.23 Example of Model Detection Failure in Scenario 1 (Dropout)	137
Figure 4.24 Stable Detection of Model Scenario 2	138
Figure 4.25 Model Detection in Scenario 2 with Position Changes	138
Figure 4.26 Detection of the letter A by the Scenario 1 model (still detectable).....	142

Figure 4.27 Detection of the letter A by the Scenario 1 model under changed lighting	142
Figure 4.28 Detection of the letter A in Model Scenario 2 (stable conditions) ..	143
Figure 4.29 Detection of the letter A in Model Scenario 2: varying lighting condition.....	143
Figure 4.30 Detection of Letter A in Scenario 1 under Normal Conditions (Unstable).....	148
Figure 4.31 Detection of the Letter A: Scenario 1 Model with Lighting Variations	149
Figure 4.32 Detection of Letter A: Scenario 2 Normal Conditions (Stable)	149
Figure 4.33 Detection of Letter A: Scenario 2 Normal Conditions (Consistent)	150
Figure 4.34 Detection of Letter A: Scenario 2 Lighting Variations (Stable).....	150
Figure 4.35 Detection of Letter A: Scenario 2 Lighting Variations (Consistent)	151

LIST OF TABLES

Table 2.1 Previous Research	8
Table 3.1 Example of Annotation Visualization.....	36
Table 3.2 Probability Distribution of Augmentation Combinations.....	48
Table 3.3 Comparison of Training vs. Testing and Validation Data Pipelines.....	49
Table 3.4 Example Pipeline Output for the Same Image.....	51
Table 3.5 Error Scenarios and Handling	52
Table 3.6 Model Summary.....	53
Table 3.7 Feature Map Sizes and Downsampling Factors of the ResNet-50 Backbone.....	56
Table 3.8 Examples of 2D Sinusoidal Positional Encoding Values for Several Tokens	59
Table 3.9 Results of the Calculation of L_{box}	74
Table 3.10 Matching Cost Calculation Results.....	75
Table 3.11 Comparison of Scenarios	86
Table 3.12 General Configuration.....	86
Table 4.1 Dataset Distribution by Class.....	92
Table 4.2 Annotation Results Statistics	93
Table 4.3 Example of Bounding Box Annotation Results for Letter A.....	93
Table 4.4 Visualization of Annotation Results	94
Table 4.5 Data Dimension Changes.....	96
Table 4.6 Visualization of Augmentation Results	98
Table 4.7 Model Training Parameters.....	99
Table 4.8 PyCOCOTools Evaluation Results for Scenario 1 Model.....	112
Table 4.9 Detection Results for the Test Data of Scenario 1	117
Table 4.10 PyCOCOTools Evaluation Results for Scenario 2 Model.....	119
Table 4.11 Detection Results of Scenario II Model Test Data with Augmentation	125
Table 4.12 Comparison of Training Characteristics Between Scenarios.....	126
Table 4.13 Comparison of PyCOCOTools Evaluation Results Across Scenarios	127
Table 4.14 Real-Time Detection Performance on CPU and GPU	135

Table 4.16 Visualization of Real-Time Detection Results Comparison Under Normal Conditions	139
Table 4.17 Robustness test results against dynamic lighting variations	144
Table 4.18 Visualization of Real-Time Detection Results Comparison Under Different Lighting Conditions.....	145
Table 4.19 Robustness Test Results for Different Subjects.....	151

LIST OF CODE PROGRAMS

Program Code 4.1 Data Acquisition Process	90
Program Code 4.2 Data Preprocessing.....	96
Program Code 4.3 On-the-fly Data Augmentation	98
Program Code 4.4 DETR Model Training.....	101
Program Code 4.5 Model Evaluation Scenario I	112
Program Code 4.6 Model Evaluation Scenario II	119
Program Code 4.78 Streamlit Inference Pipeline.....	131
Program Code 4.87 Real-time Detection	134

LIST OF APPENDICES

Appendix 1. Dataset	175
Appendix 2. Annotation Visualization.....	181
Appendix 3. Detection Results on Test Data (Scenario 1)	185
Appendix 4. Detection Results on Test Data (Scenario 2)	189
Appendix 5. Visualization of Real-Time Detection Results Comparison Under Normal Conditions	193
Appendix 6. Visualization of Real-Time Detection Results Comparison Under Different Lighting Conditions.....	203
Appendix 7. Different Subject Robustness Testing	213

LIST OF NOTATIONS

AP50	:	Average Precision at IoU threshold 0.5
AP75	:	Average Precision at IoU threshold 0.75
AP50-95	:	Average Precision at IoU threshold 0.50–0.95
IoU	:	Intersection over Union, a metric for bounding box overlap
x, y	:	Bounding box center coordinates (horizontal, vertical)
w, h	:	Bounding box width and height
x_{norm}, y_{norm}	:	Normalized bounding box center coordinates (0–1)
H, S, V	:	Hue, Saturation, Value in the HSV color space
α	:	Brightness factor in Color Jitter
β	:	Contrast factor in Color Jitter
γ	:	Saturation factor in Color Jitter
δ	:	Hue factor in Color Jitter
H, W, C	:	Image height, width, and number of channels
d	:	Latent dimension in the feature embedding (256)
N	:	Number of object queries (25 in this study)
C	:	Number of predicted classes (44 = 43 + no-object)
b_{GT}	:	Ground truth bounding box
b^i	:	Predicted bounding box from the i -th query
$p^i(c)$:	Prediction probability of the i -th query for class c
L_{box}	:	Bounding box loss
λ_{L1}	:	Weight for the L1 loss component
λ_{iou}	:	Weight for the GIoU loss component
L_{GIoU}	:	Generalized IoU loss
L_{cls}	:	Classification loss
L_{match}	:	Matching cost between ground truth and prediction
σ^{\wedge}	:	Optimal pair permutation produced by the Hungarian Algorithm