UNIVERSITEIT·STELLENBOSCH·UNIVERSITY
jou kennisvennoot • your knowledge partner

A Critical Analysis of Design Flaws in the Death Star

Luke Skywalker
99652154

Report submitted in partial fulfilment of the requirements of the module Project (E) 448 for the degree Baccalaureus in Engineering in the Department of Electrical and Electronic Engineering at Stellenbosch University.

Supervisor: Dr O. W. Kenobi

October 2099

Acknowledgements

I would like to thank my dog, Muffin. I also would like to thank the inventor of the incubator; without him/her, I would not be here. Finally, I would like to thank Dr Herman Kamper for this amazing report template.

UNIVERSITEIT•STELLENBOSCH•UNIVERSITY
jou kennisvennoot • your knowledge partner

Plagiaatverklaring / *Plagiarism Declaration*

1. Plagiaat is die oorneem en gebruik van die idees, materiaal en ander intellektuele eiendom van ander persone asof dit jou eie werk is.
   Plagiarism is the use of ideas, material and other intellectual property of another's work and to present is as my own.

2. Ek erken dat die pleeg van plagiaat 'n strafbare oortreding is aangesien dit 'n vorm van diefstal is.
   I agree that plagiarism is a punishable offence because it constitutes theft.

3. Ek verstaan ook dat direkte vertalings plagiaat is.
   I also understand that direct translations are plagiarism.

4. Dienooreenkomstig is alle aanhalings en bydraes vanuit enige bron (ingesluit die internet) volledig verwys (erken). Ek erken dat die woordelikse aanhaal van teks sonder aanhalingstekens (selfs al word die bron volledig erken) plagiaat is.
   Accordingly all quotations and contributions from any source whatsoever (including the internet) have been cited fully. I understand that the reproduction of text without quotation marks (even when the source is cited) is plagiarism

5. Ek verklaar dat die werk in hierdie skryfstuk vervat, behalwe waar anders aangedui, my eie oorspronklike werk is en dat ek dit nie vantevore in die geheel of gedeeltelik ingehandig het vir bepunting in hierdie module/werkstuk of 'n ander module/werkstuk nie.
   I declare that the work contained in this assignment, except where otherwise stated, is my original work and that I have not previously (in its entirety or in part) submitted it for grading in this module/assignment or another module/assignment.

Studentenommer / *Student number*	Handtekening / *Signature*
Voorletters en van / *Initials and surname*	Datum / *Date*

Abstract

English

The English abstract.

Afrikaans

Die Afrikaanse uittreksel.

Contents

List of Figures

List of Tables

Nomenclature

Variables and functions

$p(x)$	Probability density function with respect to variable x.
$P(A)$	Probability of event A occurring.
ε	The Bayes error.
ε_u	The Bhattacharyya bound.
B	The Bhattacharyya distance.
s	An HMM state. A subscript is used to refer to a particular state, e.g. s_i refers to the i^{th} state of an HMM.
$\mathbf{S}$	A set of HMM states.
$\mathbf{F}$	A set of frames.
$\mathbf{o}_f$	Observation (feature) vector associated with frame f.
$\gamma_s(\mathbf{o}_f)$	A posteriori probability of the observation vector $\mathbf{o}_f$ being generated by HMM state s.
μ	Statistical mean vector.
Σ	Statistical covariance matrix.
$L(\mathbf{S})$	Log likelihood of the set of HMM states $\mathbf{S}$ generating the training set observation vectors assigned to the states in that set.
$\mathcal{N}(\mathbf{x}\|\mu, \Sigma)$	Multivariate Gaussian PDF with mean μ and covariance matrix Σ.
a_{ij}	The probability of a transition from HMM state s_i to state s_j.
N	Total number of frames or number of tokens, depending on the context.
D	Number of deletion errors.
I	Number of insertion errors.
S	Number of substitution errors.

Acronyms and abbreviations

AE	Afrikaans English
AID	accent identification
ASR	automatic speech recognition
AST	African Speech Technology
CE	Cape Flats English
DCD	dialect-context-dependent
DNN	deep neural network
G2P	grapheme-to-phoneme
GMM	Gaussian mixture model
HMM	hidden Markov model
HTK	Hidden Markov Model Toolkit
IE	Indian South African English
IPA	International Phonetic Alphabet
LM	language model
LMS	language model scaling factor
MFCC	Mel-frequency cepstral coefficient
MLLR	maximum likelihood linear regression
OOV	out-of-vocabulary
PD	pronunciation dictionary
PDF	probability density function
SAE	South African English
SAMPA	Speech Assessment Methods Phonetic Alphabet

Chapter 1

Introduction

The last few years have seen great advances in speech recognition. Much of this progress is due to the resurgence of neural networks; most speech systems now rely on deep neural networks (DNNs) with millions of parameters [?, ?]. However, as the complexity of these models has grown, so has their reliance on labelled training data. Currently, system development requires large corpora of transcribed speech audio data, texts for language modelling, and pronunciation dictionaries. Despite speech applications becoming available in more languages, it is hard to imagine that resource collection at the required scale would be possible for all 7000 languages spoken in the world today.

I really like apples.

1.1. Section heading

This is some section with two table in it: Table 1.1 and Table 1.2.

Table 1.1: Performance of the unconstrained segmental Bayesian model on TIDigits1 over iterations in which the reference set is refined.

Metric	1	2	3	4	5
WER (%)	35.4	23.5	21.5	21.2	22.9
Average cluster purity (%)	86.5	89.7	89.2	88.5	86.6
Word boundary F-score (%)	70.6	72.2	71.8	70.9	69.4
Clusters covering 90% of data	20	13	13	13	13

Table 1.2: A table with an example of using multiple columns.

	Accuracy (%)		
Model	Intermediate	Output	Bitrate
Baseline	27.5	26.4	116
VQ-VAE	26.0	22.1	190
CatVAE	28.7	24.3	215

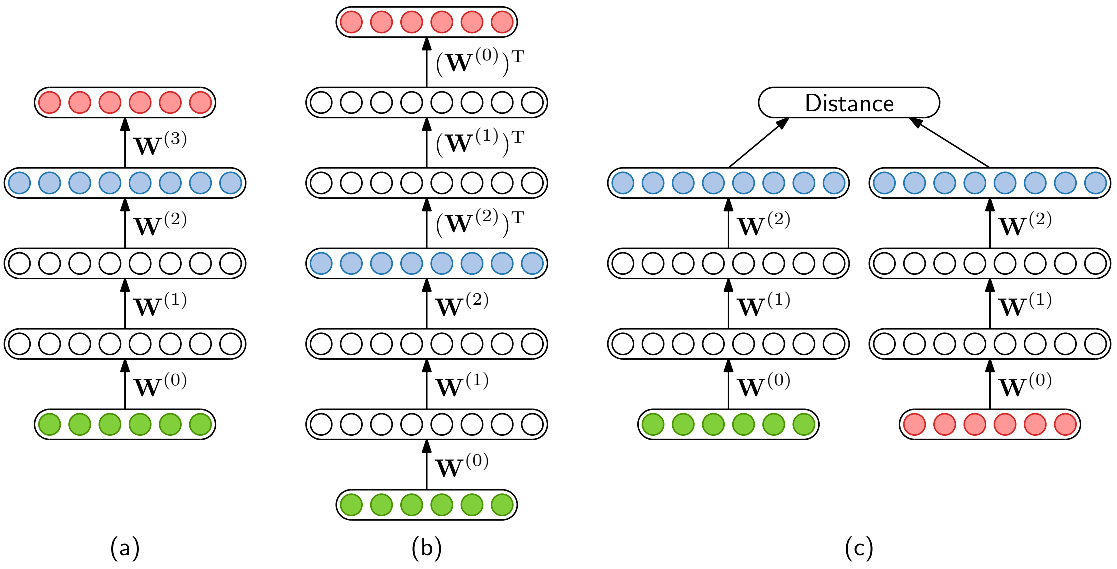

Figure 1.1: (a) The cAE as used in this chapter. The encoding layer (blue) is chosen based on performance on a development set. (b) The cAE with symmetrical tied weights. The encoding from the middle layer (blue) is always used. (c) The siamese DNN. The cosine distance between aligned frames (green and red) is either minimized or maximized depending on whether the frames belong to the same (discovered) word or not. A cAE can be seen as a type of DNN [?].

This is a new page, showing what the page headings looks like, and showing how to refer to a figure like Figure 1.1.

The following is an example of an equation:

$$P(\mathbf{z}|\boldsymbol{\alpha}) = \int_{\boldsymbol{\pi}} P(\mathbf{z}|\boldsymbol{\pi})\, p(\boldsymbol{\pi}|\boldsymbol{\alpha})\, \mathrm{d}\boldsymbol{\pi} = \int_{\boldsymbol{\pi}} \prod_{k=1}^{K} \pi_k^{N_k} \frac{1}{B(\boldsymbol{\alpha})} \prod_{k=1}^{K} \pi_k^{\alpha_k - 1}\, \mathrm{d}\boldsymbol{\pi} \tag{1.1}$$

which you can subsequently refer to as (1.1) or Equation 1.1. But make sure to consistently use the one or the other (and not mix the two ways of referring to equations).

Chapter 2

Feature Extractors

2.1. Introduction

In this report, we conduct an in-depth evaluation of various feature extractors to determine the most suitable one for real-time UAV-based navigation. Feature extractors play a pivotal role in image processing tasks, including object detection, recognition, and tracking. The chosen feature extractor must be efficient, accurate, and compatible with the high processing power capabilities of modern UAVs, ensuring reliable performance under varied conditions.

2.2. Choosing a Feature Extractor

When selecting a feature extractor, several critical factors must be considered:

- **Accuracy:** Accurate feature detection and description are crucial for navigation tasks to ensure precise localization and mapping. Inaccurate features can lead to errors in navigation and obstacle avoidance.
- **Speed:** Real-time processing necessitates fast computation and feature extraction to minimize latency, which is essential for timely decision-making in dynamic environments.
- **Robustness:** The feature extractor should exhibit invariance to changes in scale, rotation, illumination, and noise to ensure consistent performance across different flight conditions and environments.
- **Complexity:** Computational complexity directly impacts processing time and resource consumption, which is crucial for resource-constrained UAV systems where power efficiency is a consideration.
- **Scalability:** The algorithm should efficiently handle large datasets or high-resolution images, maintaining performance as the scale of data increases.

2.3. Parameters Associated with Feature Extractors

Feature extractors are characterized by several parameters that influence their performance:

2.3.1. Descriptor Size

The descriptor size, or the length of the feature vector, significantly impacts both the discriminative power and computational load of the feature extractor.

- **Implementation:** Common descriptors include SIFT (128 dimensions), SURF (64 or 128 dimensions), and ORB (32 or 256 dimensions). Larger descriptors like those used in SIFT capture more detailed information about each keypoint.
- **Pros:** Larger descriptors are more discriminative and better at distinguishing between different keypoints, improving matching accuracy across varied conditions.
- **Cons:** Increased descriptor size results in higher computational costs for storage and matching, which can be a bottleneck in real-time applications. For instance, matching thousands of 128-dimensional descriptors can be significantly slower than matching 32-dimensional descriptors.
- **Impact:** The choice of descriptor size must balance the need for detail and accuracy against the available computational resources and real-time constraints.

Mathematically, if d is the descriptor size and N is the number of keypoints, the storage requirement is $O(N \times d)$. The matching complexity, assuming a brute-force approach, is $O(N^2 \times d)$. Hence, optimizing d is crucial for real-time performance.

2.3.2. Keypoint Detection Threshold

The keypoint detection threshold determines the sensitivity of the feature detector to identifying keypoints in the image.

- **Implementation:** In algorithms like SIFT and SURF, the threshold is used to filter out weak keypoints. ORB uses the FAST detector with a threshold to select keypoints.
- **Pros:** A lower threshold increases the number of detected keypoints, which enhances robustness and ensures that important features are not missed, especially in complex or textured scenes.
- **Cons:** Increasing the number of keypoints also increases computational demands, both in terms of memory usage and processing time for descriptor computation and matching.

- **Impact:** Selecting an appropriate threshold is crucial. Too high a threshold might miss important features, reducing robustness, while too low a threshold might result in an unmanageable number of keypoints, slowing down processing.

Mathematically, if t is the threshold, then the number of keypoints N can be modeled as a function $N = f(t)$. Lowering t increases N, thus increasing the computational load $O(f(t) \times d)$.

2.3.3. Octave Layers

Octave layers refer to the levels in the scale-space pyramid used for multi-scale feature detection, enhancing robustness to scale variations.

- **Implementation:** In SIFT, the image is repeatedly blurred and subsampled to create a scale-space pyramid with multiple octaves, each containing several layers.
- **Pros:** More octave layers improve the feature extractor's ability to detect features at various scales, making it robust to changes in object size and distance.
- **Cons:** More layers increase computational complexity and processing time. Each additional octave layer involves further blurring, subsampling, and keypoint detection steps.
- **Impact:** The number of octave layers must be chosen to balance scale invariance against computational efficiency. Insufficient layers might miss features at certain scales, while too many layers could slow down processing.

Mathematically, if O is the number of octaves and L is the number of layers per octave, the computational complexity for constructing the pyramid is $O(O \times L \times N)$, where N is the number of pixels.

2.3.4. Grid Size

The grid size affects the granularity of feature extraction, impacting the detail and computational efficiency.

- **Implementation:** Grid size refers to the spatial subdivision of the image into regions for feature extraction. Smaller grid sizes provide finer granularity.
- **Pros:** Finer grids capture more detailed features and spatial relationships, improving the robustness and accuracy of the feature descriptors, particularly in textured regions.
- **Cons:** Smaller grid sizes require more computational resources for both extraction and matching, as more regions need to be processed and compared.

- **Impact:** The grid size must be carefully selected based on the complexity of the scene and the available computational resources. Finer grids are beneficial for detailed scenes but may not be practical for real-time processing on resource-limited platforms.

If G is the number of grid cells, then the computational complexity increases with G, as each cell needs to be processed independently, leading to a complexity of $O(G \times f(t) \times d)$.

2.3.5. Orientation Assignment

Orientation assignment ensures rotation invariance by assigning an orientation to each keypoint, which is crucial for consistent feature matching under different rotational perspectives.

- **Implementation:** SIFT assigns an orientation based on the gradient direction around the keypoint. ORB uses intensity centroid methods for orientation assignment.
- **Pros:** Assigning orientations to keypoints makes the descriptors invariant to image rotations, improving matching accuracy when the image or object is viewed from different angles.
- **Cons:** The orientation assignment step adds to the computational complexity and processing time. Incorrect orientation assignment can degrade matching performance.
- **Impact:** Ensuring accurate and efficient orientation assignment is vital for applications involving rotational movements. It enhances the robustness of the feature extractor but needs to be balanced against the additional computational overhead.

If θ is the orientation angle, the computation involves determining θ for each keypoint, adding a complexity of $O(N)$ where N is the number of keypoints.

2.4. Types of Feature Extractors

2.4.1. Traditional Feature Extractors

SIFT (Scale-Invariant Feature Transform)

SIFT detects and describes local features in images. It is robust to changes in scale, rotation, and illumination, making it a reliable choice for many applications. However, its computational intensity can be a drawback in real-time scenarios.

- **Advantages:** High accuracy and robustness due to its multi-scale approach and precise keypoint localization. SIFT's descriptors are highly distinctive, enabling reliable matching across different views and conditions.

- **Disadvantages:** High computational cost and slower processing speed due to the extensive keypoint detection and descriptor computation steps, making it less suitable for real-time applications.

Mathematically, SIFT's computational complexity is $O(O \times L \times N \times d)$ for the scale-space construction and $O(N \times d)$ for descriptor computation, where O is the number of octaves, L is the number of layers, N is the number of keypoints, and d is the descriptor size.

SURF (Speeded-Up Robust Features)

SURF is a faster alternative to SIFT, utilizing integral images for rapid computation of image convolutions. It offers good accuracy and robustness while being computationally more efficient than SIFT.

- **Advantages:** Faster than SIFT due to its use of Haar wavelets and integral images, providing good balance between speed and accuracy. It maintains robustness to scale and rotation changes.
- **Disadvantages:** Still relatively computationally expensive compared to simpler methods like ORB, and can be less accurate than SIFT in certain complex scenarios.

Mathematically, SURF's computational complexity is reduced to $O(O \times N \times \log N)$ for the integral image computation and $O(N \times d)$ for descriptor computation.

ORB (Oriented FAST and Rotated BRIEF)

ORB combines the FAST keypoint detector and the BRIEF descriptor, providing a highly efficient feature extraction method suitable for real-time applications. It is designed to be both fast and invariant to rotation and scale.

- **Advantages:** High speed and efficiency, making it suitable for real-time applications. ORB's binary descriptors are computationally less intensive while providing sufficient discriminative power for many tasks.
- **Disadvantages:** Lower accuracy compared to SIFT and SURF, especially in complex scenes with significant variations in lighting and scale. The binary nature of BRIEF descriptors can sometimes lead to higher false match rates.

Mathematically, ORB's complexity is $O(N \times \log N)$ for FAST keypoint detection and $O(N \times d)$ for BRIEF descriptor computation, where d is typically smaller than in SIFT or SURF.

2.4.2. Deep Learning-Based Feature Extractors

CNN-Based Extractors

Convolutional Neural Networks (CNNs) have revolutionized feature extraction by learning feature representations directly from data. Models such as VGG, ResNet, and Inception have demonstrated high accuracy and robustness in various image processing tasks.

- **Advantages:** High accuracy and the ability to learn complex and hierarchical features directly from data, enabling robust performance across diverse tasks and conditions. CNNs can adapt to specific tasks through transfer learning.
- **Disadvantages:** Requires substantial computational resources for training and inference. The training process is data-intensive, often requiring large labeled datasets to achieve optimal performance.

Mathematically, CNNs involve multiple convolutional layers with a complexity of $O(N \times K^2 \times C)$ per layer, where N is the number of pixels, K is the kernel size, and C is the number of channels. The total complexity depends on the depth and architecture of the network.

Pre-trained Models

Utilizing pre-trained models on large datasets like ImageNet can significantly expedite the development process and improve accuracy. These models can be fine-tuned for specific tasks, leveraging their learned representations.

- **Advantages:** High accuracy due to extensive pre-training on large and diverse datasets. Reduced training time and resource requirements during fine-tuning for specific applications.
- **Disadvantages:** May not generalize well to specific tasks without adequate fine-tuning. Potential for overfitting if the fine-tuning dataset is not representative of the target application.

Mathematically, the fine-tuning process involves updating a subset of weights, reducing the complexity to $O(M \times K^2 \times C)$, where M is the number of modified weights, typically much smaller than the total number of weights.

SuperPoint

SuperPoint is a self-supervised framework for interest point detection and description, employing a fully convolutional neural network to detect points and describe them in a unified process. It is designed for real-time applications and offers a balance between accuracy and efficiency.

- **Advantages:** High accuracy due to its end-to-end learning of keypoints and descriptors. It is robust to various transformations and efficient for real-time applications, leveraging deep learning techniques for feature extraction.
- **Disadvantages:** Requires substantial computational resources for training. While inference is efficient, domain-specific fine-tuning may still be necessary to achieve optimal performance in specialized tasks.

Mathematically, SuperPoint's complexity is similar to CNNs, with the additional step of keypoint extraction and description being integrated into a unified process, leading to an overall complexity of $O(N \times K^2 \times C)$ for inference.

2.5. Evaluation for UAV-Based Navigation

2.5.1. Requirements

For the Skripsie project, the feature extractor must:

- Be capable of real-time processing to ensure timely navigation and decision-making in dynamic environments.
- Utilize the available high processing power of UAV systems efficiently, balancing accuracy and speed to maintain performance without excessive computational burden.
- Provide high accuracy to ensure reliable navigation, obstacle avoidance, and mapping, which are critical for UAV operations.

2.5.2. Recommended Feature Extractor

Given the requirements of real-time processing, high accuracy, and efficient utilization of processing power, a deep learning-based feature extractor is recommended. Particularly, a pre-trained CNN model such as ResNet or VGG, or a specialized model like SuperPoint, offers the best balance of these attributes. These models leverage the high processing power of modern UAV hardware, providing robust and accurate feature extraction essential for reliable navigation.

2.6. Conclusion

In conclusion, the choice of a feature extractor depends on the specific needs of the application. For UAV-based navigation in the Skripsie project, deep learning-based extractors, especially pre-trained models and SuperPoint, offer superior performance in terms of accuracy and real-time processing capabilities. These models provide a robust solution, leveraging advanced computational resources to meet the stringent demands of UAV navigation tasks.

Chapter 3

Results

this table shows the st. dev. related to the estimation vs ground truth coordinates ratio. A more stable analysis implies a lower variation in the factor relating estimation (pixel change) to that of the true GPS coordinate change. A gaussian blur was applied with varying, square kernel sizes and the st. dev. was analysed. code as of 4aug (sift etc)

Kernel Size	Standard Deviation X	Standard Deviation Y	Norm
1	0.15985	0.03265	0.16317
3	0.16372	0.03547	0.16753
5	0.16166	0.03307	0.16401
11	0.17329	0.02227	0.17472
13	0.17310	0.03345	0.17630
15	0.17972	0.04036	0.18417
17	0.18370	0.05149	0.19177
19	0.15727	0.03997	0.16225
21	0.18859	0.03368	0.19159
23	0.15842	0.03414	0.16207
25	0.17395	0.03374	0.17720
27	0.15788	0.02341	0.15960
29	0.15515	0.03875	0.16095
41	0.15982	0.03484	0.16359
61	0.42036	0.08671	0.42918
91	1.22744	0.03476	1.22793
121	0.23488	0.12851	0.26695
181	1.22640	2.65690	2.93890

Table 3.1: Standard Deviation and Norm of the Ratio of Actual to Estimated GPS Location for Various Kernel Sizes

A kernel size of 27 resulted in the lowest normalized standard deviation. It was subsequently made the primary use. A kernel size of 11 resulted in the lowest Y deviation, while a kernel size of one resulted in a fairly low deviation and ensures no important keypoints are removed. For this reason, those three kernel sizes will be tested with in future.

Comparison of SIFT with BFMatcher and SuperPoint with LightGlue for UAV Navigation

In this analysis, we compare two feature extraction and matching techniques used for estimating GPS coordinates in UAV navigation. The first method utilizes the SIFT algorithm with BFMatcher, while the second employs SuperPoint with LightGlue. Both methods were applied to a dataset of aerial images, and the deviations between actual and estimated GPS coordinates were calculated. It's worth noting that the parameters for both methods were not optimized, which might affect their performance. The table below summarizes the results, showing the deviations in meters and the Euclidean norm for each method.

Image	**Deviation X (m)**	**Deviation Y (m)**	**Norm (m)**
SIFT + BFMatcher			
10	53.62	16.81	56.15
9	1287.48	5.02	1287.49
8	24.02	4.42	24.42
7	2.33	0.62	2.42
6	452.92	27.80	453.77
5	58.87	28.16	65.29
4	28.74	86.32	90.82
3	7.51	14.52	16.31
2	24.96	16.10	29.64
SuperPoint + LightGlue			
10	45.38	2.83	45.47
9	1281.47	41.44	1282.15
8	23.98	6.96	24.95
7	3.48	0.80	3.57
6	452.04	15.92	452.34
5	58.04	27.70	64.24
4	15.15	63.60	65.37
3	2.67	18.44	18.64
2	21.39	20.56	29.59

Table 3.2: Comparison of Deviation and Norm for SIFT + BFMatcher and SuperPoint + LightGlue

Analysis

In this comparison, we observe that the SuperPoint + LightGlue combination provides lower deviation norms in 6 out of 9 cases for the X deviations, suggesting better performance in those instances. The unoptimized parameters may have affected the overall performance, but the results indicate that SuperPoint + LightGlue could offer more accurate estimates under the same conditions.

Percentile Outliers removal

Lower Cutoff (%)	Normalized Error
0	59.57
1	58.61
2	57.71
3	57.01
4	55.33
6	55.80
10	54.96
12	54.06
14	54.00
16	54.11
18	53.98
19	53.45
20	53.82
21	54.07
22	54.44
24	54.06
26	54.00
28	54.44
30	53.66
32	54.39
34	54.14
36	54.01
38	53.49
40	53.20
41	53.46
42	53.68
43	53.82
44	53.30
45	52.87
46	53.25
47	52.48
48	52.73
49	53.74

Table 3.3: Normalized Error vs. Lower Percentile Cutoff

Chapter 4

Summary and Conclusion

Appendix A

Project Planning Schedule

This is an appendix.

Appendix B

Outcomes Compliance

This is another appendix.