Machine Learning Engineer Nanodegree

Capstone Project

Daniel Vargas
October 21, 2019

I. Definition

(approx. 1-2 pages)

Project Overview

In this section, look to provide a high-level overview of the project in layman's terms. Questions to ask yourself when writing this section:

- *Has an overview of the project been provided, such as the problem domain, project origin, and related datasets or input data?*
- *Has enough background information been given so that an uninformed reader would understand the problem domain and following problem statement?*

Facial Key Points (FKPs) detection is an important and challenging problem in the field of computer vision, which involves detecting FKPs like centers and corners of eyes, nose tip, etc. The problem is to predict the (x, y) real-valued co-ordinates in the space of image pixels of the FKPs for a given face image. It finds its application in tracking faces in images and videos, analysis of facial expressions, detection of dysmorphic facial signs for medical diagnosis, face recognition, etc. Facial features vary greatly from one individual to another, and even for a single individual there is a large amount of variation due to pose, size, position, etc. The problem becomes even more challenging when the face images are taken under different illumination conditions, viewing angles, etc. In the past few years, advancements in FKPs detection are made by the application of deep convolutional neural network (DCNN), which is a special type of feed-forward neural network with shared weights and local connectivity. DCNNs have helped build state-of-the-art models for image recognition, recommender systems, natural language processing, etc. Krizhevsky et al. [1] applied DCNN in ImageNet image classification challenge and outperformed the previous state-of-the-art model for image classification. Wang et al. [2] addressed FKPs detection by first applying histogram stretching for image contrast enhancement, followed by principal component analysis for noise reduction and mean patch search algorithm with correlation scoring and mutual information scoring for predicting left and right eye centers. Sun et al. [3] estimated FKPs by using a three level convolutional neural network, where at each level, outputs of multiple networks were fused for robust and accurate estimation. Longpre et al. [4] predicted FKPs by first applying data augmentation to expand the number of training examples, followed by testing different architectures of convolutional neural networks like LeNet [5] and VGGNet [6], and finally used a weighted ensemble of models. Nouri et al. [7] used six specialist DCNNs trained over pre-trained weights. Oneto et al. [8] applied a variety of data pre-processing techniques like histogram stretching, Gaussian blurring, followed by image flipping, key point grouping, and then finally applied LeNet. Taigman et al. [9] provided a new deep network architecture, DeepFace, for state-of-the-art face recognition. Li et al. [10] provided a new DCNN architecture for state-of-the art face alignment. We present a DCNN architecture – NaimishNet, based on LeNet, which addresses the problem of facial key points detection by providing a learning model for a single facial key point.

We have used the dataset from Kaggle Competition – Facial Key Points Detection [17]. The dataset was chosen to benchmark our solution against the existing approaches which address FKPs detection problem. There are 15 FKPs per face image like left eye center, right eye center, left eye inner corner, left eye outer corner, right eye inner corner, right eye outer corner, left eyebrow inner end, left eyebrow outer end, right eyebrow inner end, right eyebrow outer end, nose tip, mouth left corner, mouth right corner, mouth center top lip and mouth center bottom lip. Here, left and right are from the point of view of the subject. The greyscale input images, with pixel values in range of [0, 255], have size of 96 x 96 pixels. The train dataset consists of 7049 images, with 30 targets, i.e. (x, y) for each of 15 FKPs. It has missing target values for many FKPs for many face images. The test dataset consists of 1783 images with no target information. The Kaggle submission file consists of 27124 FKPs co-ordinates, which are to be predicted.

Problem Statement

In this section, you will want to clearly define the problem that you are trying to solve, including the strategy (outline of tasks) you will use to achieve the desired solution. You should also thoroughly discuss what the intended solution will be for this problem. Questions to ask yourself when writing this section:

- *Is the problem statement clearly defined? Will the reader understand what you are expecting to solve?*
- *Have you thoroughly discussed how you will attempt to solve the problem?*
- *Is an anticipated solution clearly defined? Will the reader understand what results you are looking for?*

Metrics

In this section, you will need to clearly define the metrics or calculations you will use to measure performance of a model or result in your project. These calculations and metrics should be justified based on the characteristics of the problem and problem domain. Questions to ask yourself when writing this section:

- *Are the metrics you've chosen to measure the performance of your models clearly discussed and defined?*
- *Have you provided reasonable justification for the metrics chosen based on the problem and solution?*

RMSE formula here

II. Analysis

(approx. 2-4 pages)

Data Exploration

In this section, you will be expected to analyze the data you are using for the problem. This data can either be in the form of a dataset (or datasets), input data (or input files), or even an environment. The type of data should be thoroughly described and, if possible, have basic statistics and information presented (such as discussion of input features or defining characteristics about the input or environment). Any abnormalities or interesting qualities about the data that may need to be addressed have been identified (such as features that need to be transformed or the possibility of outliers). Questions to ask yourself when writing this section:

- *If a dataset is present for this problem, have you thoroughly discussed certain features about the dataset? Has a data sample been provided to the reader?*
- *If a dataset is present for this problem, are statistics about the dataset calculated and reported? Have any relevant results from this calculation been discussed?*

- *If a dataset is **not** present for this problem, has discussion been made about the input space or input data for your problem?*
- *Are there any abnormalities or characteristics about the input space or dataset that need to be addressed? (categorical variables, missing values, outliers, etc.)*

Data Augmentation helps boost the performance of a deep learning model when the training data is limited by generating more training data. We have horizontally flipped [7] the images for which target information for all the 15 FKPs are available, and also swapped the target values according to Table 3. Then, we vertically stacked the new horizontally flipped data under the original train data to create the augmented train dataset.

Data Preprocessing

The image pixels are normalized to the range [0, 1] by dividing by 255.0, and the train targets are zero-centered to the range [-1, 1] by first dividing by 48.0, since the images are 96 x 96, and then subtracting 48.0.

Figure 1 shows that there are different number of non-missing target rows for different FKPs, so, we have created 15 NaimishNet models.

Exploratory Visualization

In this section, you will need to provide some form of visualization that summarizes or extracts a relevant characteristic or feature about the data. The visualization should adequately support the data being used. Discuss why this visualization was chosen and how it is relevant. Questions to ask yourself when writing this section:

- *Have you visualized a relevant characteristic or feature about the dataset or input data?*
- *Is the visualization thoroughly analyzed and discussed?*
- *If a plot is provided, are the axes, title, and datum clearly defined?*

Algorithms and Techniques

In this section, you will need to discuss the algorithms and techniques you intend to use for solving the problem. You should justify the use of each one based on the characteristics of the problem and the problem domain. Questions to ask yourself when writing this section:

- *Are the algorithms you will use, including any default variables/parameters in the project clearly defined?*
- *Are the techniques to be used thoroughly discussed and justified?*
- *Is it made clear how the input data or datasets will be handled by the algorithms and techniques chosen?*

Benchmark

In this section, you will need to provide a clearly defined benchmark result or threshold for comparing across performances obtained by your solution. The reasoning behind the benchmark (in the case where it is not an established result) should be discussed. Questions to ask yourself when writing this section:

- *Has some result or value been provided that acts as a benchmark for measuring performance?*
- *Is it clear how this result or value was obtained (whether by data or by hypothesis)?*

III. Methodology

(approx. 3-5 pages)

Data Preprocessing

In this section, all of your preprocessing steps will need to be clearly documented, if any were necessary. From the previous section, any of the abnormalities or characteristics that you identified about the dataset will be addressed and corrected here. Questions to ask yourself when writing this section:

- *If the algorithms chosen require preprocessing steps like feature selection or feature transformations, have they been properly documented?*
- *Based on the **Data Exploration** section, if there were abnormalities or characteristics that needed to be addressed, have they been properly corrected?*
- *If no preprocessing is needed, has it been made clear why?*

Implementation

In this section, the process for which metrics, algorithms, and techniques that you implemented for the given data will need to be clearly documented. It should be abundantly clear how the implementation was carried out, and discussion should be made regarding any complications that occurred during this process. Questions to ask yourself when writing this section:

- *Is it made clear how the algorithms and techniques were implemented with the given datasets or input data?*
- *Were there any complications with the original metrics or techniques that required changing prior to acquiring a solution?*
- *Was there any part of the coding process (e.g., writing complicated functions) that should be documented?*

Refinement

In this section, you will need to discuss the process of improvement you made upon the algorithms and techniques you used in your implementation. For example, adjusting parameters for certain models to acquire improved solutions would fall under the refinement category. Your initial and final solutions should be reported, as well as any significant intermediate results as necessary. Questions to ask yourself when writing this section:

- *Has an initial solution been found and clearly reported?*
- *Is the process of improvement clearly documented, such as what techniques were used?*
- *Are intermediate and final solutions clearly reported as the process is improved?*

IV. Results

(approx. 2-3 pages)

Model Evaluation and Validation

In this section, the final model and any supporting qualities should be evaluated in detail. It should be clear how the final model was derived and why this model was chosen. In addition, some type of analysis should be used to validate the robustness of this model and its solution, such as manipulating the input data or environment to see how the model's solution is affected (this is called sensitivity analysis). Questions to ask yourself when writing this section:

- *Is the final model reasonable and aligning with solution expectations? Are the final parameters of the model appropriate?*
- *Has the final model been tested with various inputs to evaluate whether the model generalizes well to unseen data?*
- *Is the model robust enough for the problem? Do small perturbations (changes) in training data or the input space greatly affect the results?*
- *Can results found from the model be trusted?*

Justification

In this section, your model's final solution and its results should be compared to the benchmark you established earlier in the project using some type of statistical analysis. You should also justify whether these results and the solution are significant enough to have solved the problem posed in the project. Questions to ask yourself when writing this section:

- *Are the final results found stronger than the benchmark result reported earlier?*
- *Have you thoroughly analyzed and discussed the final solution?*
- *Is the final solution significant enough to have solved the problem?*

V. Conclusion

(approx. 1-2 pages)

Free-Form Visualization

In this section, you will need to provide some form of visualization that emphasizes an important quality about the project. It is much more free-form, but should reasonably support a significant result or characteristic about the problem that you want to discuss. Questions to ask yourself when writing this section:

- *Have you visualized a relevant or important quality about the problem, dataset, input data, or results?*
- *Is the visualization thoroughly analyzed and discussed?*
- *If a plot is provided, are the axes, title, and datum clearly defined?*

Reflection

In this section, you will summarize the entire end-to-end problem solution and discuss one or two particular aspects of the project you found interesting or difficult. You are expected to reflect on the project as a whole to show that you have a firm understanding of the entire process employed in your work. Questions to ask yourself when writing this section:

- *Have you thoroughly summarized the entire process you used for this project?*
- *Were there any interesting aspects of the project?*
- *Were there any difficult aspects of the project?*
- *Does the final model and solution fit your expectations for the problem, and should it be used in a general setting to solve these types of problems?*

Improvement

In this section, you will need to provide discussion as to how one aspect of the implementation you designed could be improved. As an example, consider ways your implementation can be made more general, and what would need to be modified. You do not need to make this improvement, but the potential solutions resulting from these changes are considered and compared/contrasted to your current solution. Questions to ask yourself when writing this section:

- *Are there further improvements that could be made on the algorithms or techniques you used in this project?*
- *Were there algorithms or techniques you researched that you did not know how to implement, but would consider using if you knew how?*
- *If you used your final solution as the new benchmark, do you think an even better solution exists?*

Before submitting, ask yourself. . .

- Does the project report you've written follow a well-organized structure similar to that of the project template?
- Is each section (particularly **Analysis** and **Methodology**) written in a clear, concise and specific fashion? Are there any ambiguous terms or phrases that need clarification?
- Would the intended audience of your project be able to understand your analysis, methods, and results?
- Have you properly proof-read your project report to assure there are minimal grammatical and spelling mistakes?
- Are all the resources used for this project correctly cited and referenced?
- Is the code that implements your solution easily readable and properly commented?
- Does the code execute without error and produce results similar to those reported?