

# Gaussian Process Guided Neural Networks: An Exploration

Jake Callahan, Taylor Paskett

March 20, 2021

Things to talk about:

- What does Gaussian process bring to the table that regular NN's lack?
- Why is using a GPGNN a good idea? What does it do to the model? i.e. a PGNN helps ensure results stay physically consistent
- data
- Model architecture: based off Whenet, etc
- novel loss function
- how we trained
- results

## 1 Introduction

A major problem facing neural networks today is the size of their required training set. Neural networks need a large amount of data to train an accurate model. Further, neural networks can only learn relationships that exist in these data, and aren't well-suited to ingesting and learning from prior knowledge about the data.

Conversely, many Bayesian inference- based statistical methods are well-suited for training with smaller amounts of data and rely on imposed prior knowledge. However, they suffer from infeasibility at large scales and lack versatility across problem types.

We propose a network architecture that combines the strengths of these models and mitigates their weaknesses. Inspired by physics-guided neural networks (PGNN), which rely on physical models to guide the training of the neural network [1], we create a similar that uses a Gaussian process to guide the training of a conventional neural network. **CONFIRM: In this paper, we show that on a face-pose detection problem, the Gaussian Process Guided Neural Network (GPNN) achieves similar results to the current state-of-the-art [3] while using less data.**

## 2 Regular model vs. GPGNN

## 3 Data

We apply our GPGNN model to a common problem: facial pose detection. Specifically, we use the 300W-LP dataset [4]. This dataset is composed of one input variable, a close-up facial image, and three outcome variables which define the facial pose: pitch, yaw, and roll. These pose parameters are given in degrees, and range from  $-90^\circ$  to  $90^\circ$ . This dataset is widely used in face-pose estimation problems and as such is a natural choice for training the GPGNN.

## 4 Architecture

### 4.1 WHENet

We base our model architecture on Zhou and Gregson's WHENet model [3]. This model is the current state-of-the art, and will be used as a baseline comparison for the success of our GPGNN. WHENet is an extension of the successful EfficientNet model developed by Tan and Le [2]. WHENet uses EfficientNet as its first layer. It then takes the EfficientNet outputs and applies two separate loss functions to each of the yaw, pitch, and roll.

One loss function is a mean-squared error loss that measures the model's regression performance, and the other is a cross-entropy loss that measures how well the model would do if it were a classification problem. This cross-entropy loss is applied by first binning each of the yaw, pitch and roll into 60 bins of 3 degrees each, and then feeding these bins to the cross-entropy loss function. The total loss of the model is the sum across all loss terms. See Figure 1 (courtesy of [3]) for a visualization of this architecture.

### 4.2 GPGNN

Our GPGNN architecture is highly similar. We

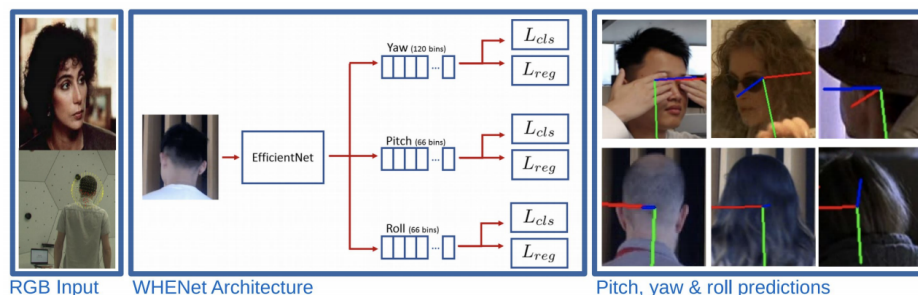


Figure 1: Illustration of WHENet architecture

## References

- [1] Anuj Karpatne, William Watkins, Jordan S. Read, and Vipin Kumar. Physics-guided neural networks (PGNN): an application in lake temperature modeling. *CoRR*, abs/1710.11431, 2017. URL <http://arxiv.org/abs/1710.11431>.
- [2] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.
- [3] Yijun Zhou and James Gregson. Whenet: Real-time fine-grained estimation for wide range head pose. 2020. URL <https://arxiv.org/pdf/2005.10353.pdf>.
- [4] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z. Li. Face alignment in full pose range: A 3d total solution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):78–92, Jan 2019. ISSN 1939-3539. doi: 10.1109/tpami.2017.2778152. URL <http://dx.doi.org/10.1109/TPAMI.2017.2778152>.