Ground Slope Prediction Using Convolutional Neural Networks on Smartphone Motion Data

Merha Berihe, Shu Naito, Sahith Desham, Jason Xie

***Abstract*— Smartphones have long been used to track users' health through metrics such as step count and walking distance. More recently, advances in machine learning have enabled even more nuanced health metrics to be measured, such as gait symmetry. However, most gait tracking methods rely on the assumption that the ground is flat, which leads to reduced reliability when the ground is inclined. In this paper, we propose the next step forward in everyday health tracking by using convolutional neural networks to predict ground slope during walking. The network employs parallel, multi-scale 1D convolutions and residual connections to extract features from acceleration, angular velocity, and orientation data collected with a phone placed in the pant pocket. We compare the results using our deep neural network with that of logistic regression, highlighting the fact that logistic regression can be combined with the continuous wavelet transform to make predictions on temporal data and inform feature selection.**

I. INTRODUCTION

Ground slope prediction is the process of measuring the incline at which a person walks or a machine moves. Although measuring slope is usually taken for granted using visual observations or the use of a level, it is to be seen whether the ground slope of a walking human or a biped robot can be measured through non-stationary sources of data. Intuition suggests that prediction would be possible. After all, a walking human being could adequately guess the incline of the ground-based on the amount of effort needed to propel oneself forward.

The measurement of slope using non-stationary data would have many benefits. One of the most significant impacts would be in the field of personalized health and fitness. With the advent and growth of data analytics in health, one area which has been gaining a lot of traction is health tracking. Smartphones today are capable of tracking the users' health through various metrics, such as heart rate, calories burned, and sleeping schedule. Gait analysis is also possible through the smartphone's accelerometer and gyroscope. In fact, apps that detect stride length and gait asymmetry are readily available, leading to a highly detailed analysis of a person's biomechanics. However, most analyses assume the user is walking on flat ground. Thus, their application towards walking on the sloped ground may be limited. The need for better gait analysis on the sloped ground also implies the need for slope prediction. Research has also shown that for each 1 percent increase in slope, a person with 150 lbs of weight burns around 10 more calories per mile [1]. However, the heart rate of a person also increases with slope inclines. To maximize the calorie-burn while keeping the heart rate in a risk-less zone, measurement of both the heart rate and slope is needed to find the range of slope inclines that is safe for a specific person. While it is conceivable that the ground slope can be directly calculated using height changes from a barometer and walking speed, this technique could not be applied for walking on a treadmill. Thus, ground slope prediction could instead be done using smartphone motion data, leading to potential health gains.

The authors are with the Computer Science Department at the University of North Carolina at Chapel Hill (correspondence e-mail: `jasonxie@live.unc.edu`)

Furthermore, the use of robots to reach places humans could not access is accelerating. Rescue robots can travel to places that might be unsafe to rescue workers in time of natural disasters. Robots also travel to different planets for gathering samples for research. Although wireless cameras are used by operators to see far away objects, slope measurements used to keep the balance of robots requires sensing by the wheels or legs of the robots [2].

Despite the importance of slope prediction, there are not enough research and models to estimate slopes using non-stationary gait qualities. In this study, we aim to take the next step in gait analysis by predicting ground slope using smartphone motion data. we tried to predict the slope of a walking area using angular acceleration, angular velocity, and orientation data. Using these features, we used multinomial logistic regression and a convolutional neural network to predict eight different slope values.

II. RELATED WORK

Similar work of gait mode detection has been studied in the medical field. In Islam et al., gait measurements were taken by prosthetic devices attached to the ankle area. Although gait mode detection using decision tree algorithms was ineffective, artificial neural network (ANN) algorithms were effective in classifying gait modes of level ground, ascent, and descent. [3]. Inertial measurements, vertical velocity, and segment angle of the foot were used as the features. After the inputs were run through a multi-layer feed-forward network, the output was filtered for classification. Although ANN improved the timely determination of slope changes, specific slope angle determination has remained explored. In addition, features like foot angle are limited in their applications as it will be hard to implement the same model to wheels, robots, or other animated objects. Therefore, the use of body features like orientation, velocity, and acceleration would facilitate easier generalization to different machines in addition to humans.

III. METHODS

A. Data Collection

1) Sensors: A smartphone was placed in the right pant pocket of one healthy participant to collect accelerometer and gyroscope data during the walk. The phone is oriented upside-down, with the display facing away from the body. Two rubber bands wrap the phone tightly against the thigh to prevent extraneous motion from injecting noise into data (Figure 1). During the walk, acceleration, angular velocity, and orientation were recorded, resulting in a feature with nine dimensions. Meanwhile, a tablet is placed on the floor of the treadmill to measure the ground slope. Both devices collected data at 100Hz using the MATLAB mobile app. When the walk data is combined with the slope, the resultant data is an Nx10 array, with the first 9 columns being the features, and the last column being the slope.

2) Variable Slope Track: After starting simultaneous data collection on both the user-attached smartphone and the slope tracking tablet, the participant engages in multiple walking sessions where the incline of the treadmill is incrementally changed. The participant begins by walking at a steady pace of 3 m.p.h. in the flat setting. After 3 minutes of habituation, the participant begins data collection. Two separate methods were used for collecting train/validation and test data. For the train/validation data, the participant first walks on the flat surface for another 8 minutes for data collection. Thereafter, the incline of the treadmill is raised by 1 deg every 2 minutes until the highest incline is reached. Upon reaching the top, the incline is then lowed by 1 deg every 2 minutes until the participant is again walking on the flat surface. A total of eight slope categories were present in the train/validation set. To collect the test data, the slope was incremented by 0.4 deg every 3 seconds until reaching the highest level, and then decreasing by 0.4 deg every 3 seconds until reaching the flat ground. Thus, the test set labels provided more continuous label values compared to the train/validation set.

B. Logistic Regression Using Scaleogram Features

Logistic regression was first used to establish baseline performance and identify the best features. Because the data collected is temporal in nature, plain logistic regression would be inadequate because it only uses features one sample at a time. To incorporate temporal information for logistic regression, the Continuous Wavelet Transform (CWT) was used to convert information into the frequency domain [4]. The CWT uses a basis set of wavelets to identify transient oscillatory features, thus simultaneously capturing local spectral and temporal information. The basis set is formed from the mother wavelet $\psi(t)$, with properties:

$$\int_{-\infty}^{\infty} \psi(t)dt = 0 \tag{1}$$

$$\|\psi(t)\|^2 = \int_{-\infty}^{\infty} \psi(t)\psi^*(t)dt = 0 \tag{2}$$

The basis set is denoted as:

$$\left\{ \psi_{s,\tau}(t) = \frac{1}{\sqrt(s)}\psi(\frac{t-\tau}{s}) \right\} \tag{3}$$

Where τ is the translating parameter, and s is the nonzero scaling parameter used to construct wavelets of different widths. Given a signal $f(t)$, the CWT returns a time-frequency array of wavelet coefficients given by:

$$\begin{aligned} Wf(s,\tau) &=< f(t), \psi_{s,\tau} > \\ &= \int_{-\infty}^{\infty} f(t)\psi^*_{s,\tau}(t)dt \\ &= \int_{-\infty}^{\infty} f(t)\frac{1}{\sqrt(s)}\psi^*(\frac{t-\tau}{s})dt \end{aligned}$$

Using the CWT and appropriate wavelet scales for the complex Mortlet wavelet, we computed the scaleograms for all nine features of the walking data. To facilitate classification, data labels were discretized into 8 categories. Scaleograms were normalized using the mean and standard deviation of the train/validation set scaleogram. To find the optimal set of features, we trained logistic regression separately on each of the nine features after applying the wavelet transform. Because the class labels are numerically related, root-mean-squared-error (RMSE) was used as the performance metric. Test error was found to be lowest in y-acceleration, x-angular velocity, and pitch (Table I). Thus, y-acceleration, x-angular velocity, and pitch were used for training the full logistic regression model and future neural network models. For full logistic regression on the three selected features, we first computed the scaleograms for each feature, and then concatenated the scaleograms along the frequency axis.

TABLE I
LOGISTIC REGRESSION RMSE USING INDIVIDUAL FEATURES

Feature	Train	Test
x-Acceleration	1.87	2.58
y-Acceleration	0.69	1.14
z-Acceleration	0.97	1.43
x-Angular Velocity	0.92	1.26
y-Angular Velocity	2.20	2.57
z-Angular Velocity	2.19	2.83
Azimuth	1.80	2.84
Pitch	0.65	1.18
Roll	2.00	2.68

C. 1D Convolutional Network

1) Network Architecture: To further increase the accuracy of predictions, we proposed the use of a 1D convolutional neural network (CNN) for slope prediction. The network architecture is presented in Figure 3. A prediction of the ground slope is made after the network uptakes N samples of data through a series of temporal convolutions. Inspired by deep residual networks and parallel convolution architectures

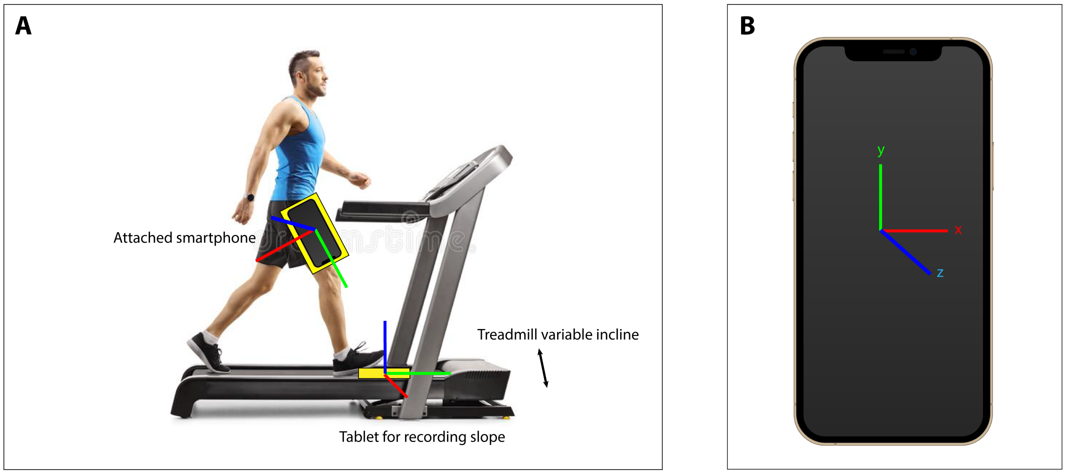

Fig. 1. Data Collection Overview. **A**: Collection setup. During collection, the participant attaches the phone to the thigh using rubber bands. Note that the phone's outward-facing depiction is for image clarity only. The phone screen should be facing the direction of travel. On the treadmill floor, a tablet records the ground slope. **B**: Device measurement frame of reference. x, y, z measurements are defined according to this frame. For example, when the phone is placed vertically as shown in the figure, there would be a y acceleration of $+9.81m/s^2$.

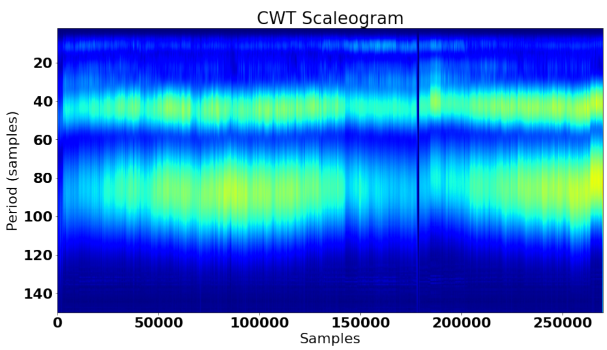

Fig. 2. Example scaleogram computed from y-acceleration.

for image classification [5] [6], we designed a parallel convolutional residual block (PConvRes) that use multi-resolution filters. The use of multi-resolution filters within the blocks would help the network become invariant to time scaling, thus leading to reliable predictions at various walking speeds. When inputs enter the PConvRes block, it is fed into three different branches. The first and second branch performs a convolution in between 1x1 convolutions that adjust the filter sizes to improve computational efficiency. Following convolutions, the first and second branches are merged with a filter-wise concatenation. The third branch is a residual connection that joins the other branches via element-wise addition. Two convolutional layers are used before PConvRes Blocks, and 3 fully connected layers are used after the PConvRes blocks.

2) Data Preparation: To prepare the data for training a neural network, we first down-sampled the data from 100Hz to 33.33Hz. Due to limitations in the amount of data collected, we sought to increase the amount of walking data available for training by further up/down-sampling the data, resulting in data that resemble walking at various speeds. Additionally, Data normalization was performed on the dataset by centering and re-scaling the data by the mean and standard deviation of the training dataset.

While the input data is a continuous stream of data, the network must accept windowed values. We define a window as a contiguous collection of data samples containing features and labels. To obtain the best window width, we trained small convolutional networks with one convolutional layer and one densely connected layer on inputs with [16, 84, 128, 300] samples for width. It was found that the 84 wide window produced the lowest validation error. Thus, the network input has dimension (batch size, 84, 3). We took the slope label of the last sample as the target. While the network makes a prediction every 84 samples, the input window shape does not need to conform to the width of the network input because the network acts as a sliding window along the input width. For example, a window can be defined to be 1000 samples wide with (1000-84+1) labels instead of 1 label.

3) Training: The training was done on a Google Colab GPU instance with a batch size of 48 and using the Adam optimizer with a learning rate of 5e-5 and the mean absolute error loss [7]. Regularization was applied to the fully-collected layers using L2 regularization and dropout at a rate of 0.5 [8].

IV. RESULTS

The movement of one subject on a treadmill was measured, and the different predictive parameters were measured. The data collected was divided into modeling and testing data, and the results were predicted based on cross-validation of different machine learning models. We utilized Root Mean Squared Error of slope predicted vs the actual slope to signal which model gave the best results.

1) Logistic Regression Using Combined Scaleogram Features: After determining that the optimal features for predicting ground slope were y-acceleration, x-angular velocity, and pitch, we trained a logistic regression model with the combined scaleograms of the optimal features. The model

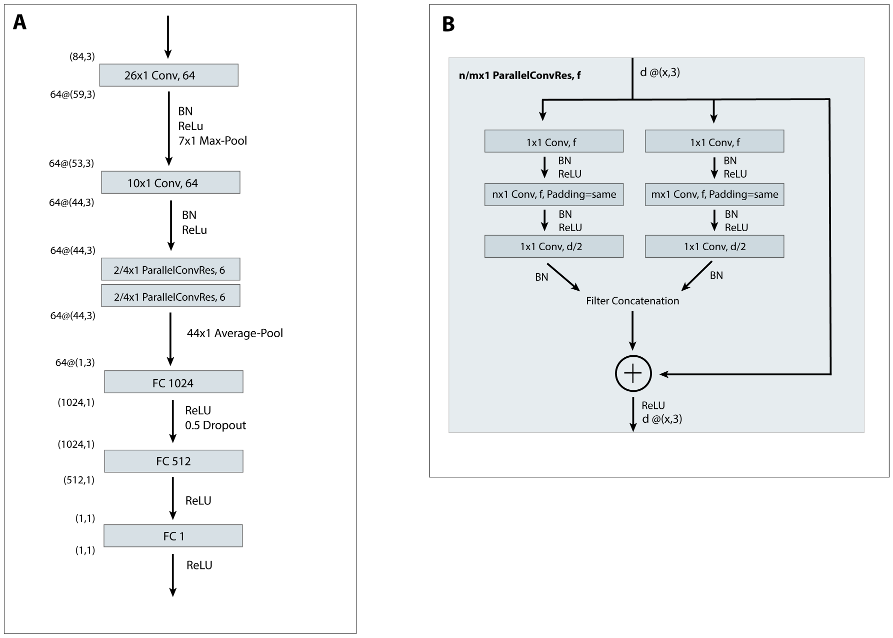

Fig. 3. Network Architecture. **A**: Main network. The network takes in 84 samples of walking data consisting of y-acceleration, x-angular velocity, and pitch, and predicts the ground slope. **B**: PConvRes block. The block splits the input into three streams, with the first and second computing convolutions with different kernal sizes.

achieved a training RMSE of 0.46 degrees and a testing RMSE of 1.01 degrees (Figure 4). The error was lowest in the lowest slope value and the highest slope value. The confusion matrix in Figure 4 shows that the logistic regression model tends to underestimate the slope. For example, the model assigned the 2nd slope class features to the first slope class $> 95\%$ of the time. The variance of prediction is also high in the test set predictions. Although class prediction accuracy was low, inaccurate predictions generally landed near the true value, suggesting that lowering the number of slope classes would increase accuracy.

2) CNN: We trained the CNN using the collected train/validation set. New predictions were made on the previously unseen test set with more continuous slope values. The train/validation RMSE history can be seen in Figure 7. The history plot indicates that the model did not over-fit the data despite its limited quantity. The lowest validation RMSE achieved was 0.56 degrees, while the test RMSE was 0.57 degrees. Despite being trained using limited label values, Figure 6 shows that the CNN adequately predicted the test set's continuous slope track.

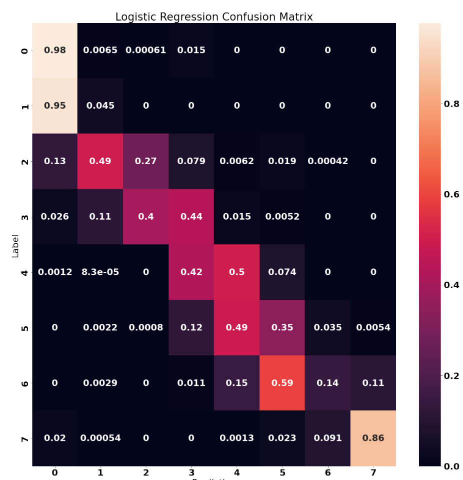

Fig. 4. Logistic Regression Confusion Matrix. It is apparent that logistic regression predictions tend to be lower than actual labels.

V. Discussion

1) Implications of Results: In our work, we demonstrated the use of a convolutional neural network to predict ground slope during walking. The model achieved good results on the test set, predicting the slope with an RMSE of 0.57 degrees. While discrete slope levels were used during training, the CNN successfully generalized to continuous slopes, leading to accurate predictions on the test set. Binning the predictions and constructing a confusion matrix shows that the model performed better for some slope levels than others (Figure 5). This may be attributed to an imbalance of slopes in the training data. During data collection, the amount of time spent on each slope was not strictly controlled. Additionally, the participant did not spend as much time on the highest slope levels due to fatigue. This may have contributed to under-prediction in the topmost values. Another observation of the predicted slopes is that slope prediction was noisy in both the CNN and logistic regression. Thus, further accuracy improvements could be attained by filtering or averaging the predictions with the cost of decreased temporal resolution.

2) Feature Choice: Before training the CNN, we had used logistic regression to evaluate performance on each of the

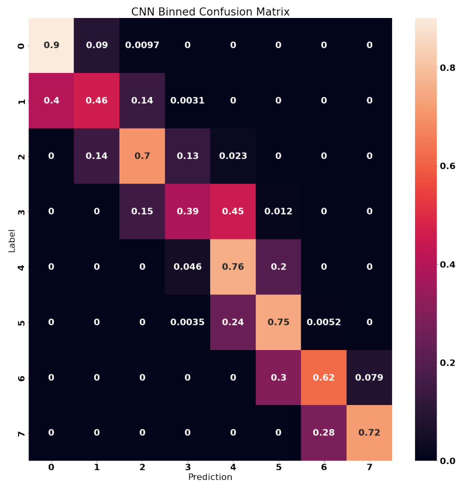

Fig. 5. CNN Binned Confusion Matrix.

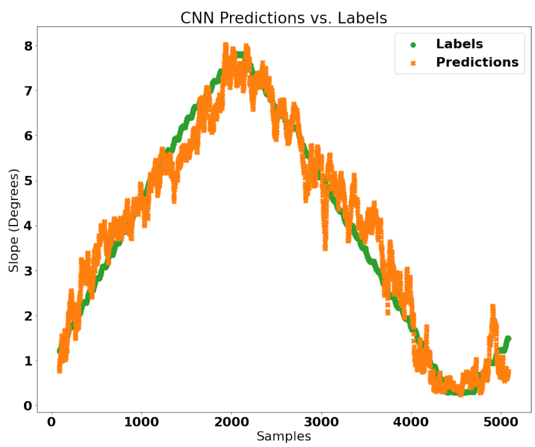

Fig. 6. Example test set predicted slope track for CNN

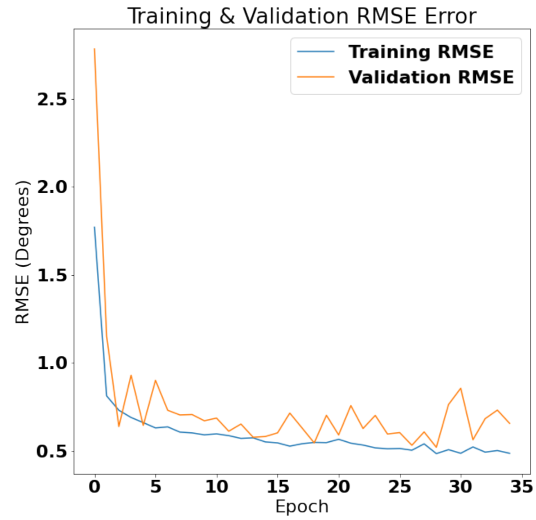

Fig. 7. Train/validation error history for CNN

nine features in the raw dataset. The three best features were determined to be y-acceleration, x-angular velocity, and pitch. While we had systematically determined their predictive power, we can also deduce the use of these three features by eliminating features that are uninformative during walking. For example, in our data collection configuration, the x-acceleration measures sideways leg sway, which is unlikely to change under different slopes. Therefore, we can speculate that x-acceleration would be not useful towards slope prediction during straight-line walking.

While we used the wavelet transform to capture temporal information for logistic regression, we did not use scale-ograms as input features for training the convolution neural network. To capture temporal information, 1D CNNs learn temporal filters that detect salient features over the course of training. This is in contrast to the wavelet transform, which used a pre-defined wavelet set that filters the data at various time scales.

3) Limitations and Future Work: To our best knowledge, this is the first time ground slope has been predicted from smartphone motion data. Our convolutional neural network has shown to be promising in the application, with an approximate 1.8x improvement over logistic regression using scaleogram features. Overall, we have shown that ground slope can be predicted using motion data from a smartphone held in a user's pocket. Our findings may lead to better health tracking systems that take ground slope into account during gait analysis, or control systems for bipedal robots that adapt its gait to various terrain conditions.

In the future, we expect ground slope prediction can be improved upon by increasing the dataset size. A large number of participants could potentially increase the robustness of the model by extracting features that are invariant across different gait styles. In our model, we only considered one phone carrying configuration. Training data with more configurations can be attained by either transforming current data, or by collecting new data by attaching phones to the participant at different orientations. It may be also possible to use sensor fusion to compute motion feature vectors that are invariant to the phone's orientation. Ultimately, obtaining more data could pave the way for larger models, thus increasing accuracy and reliability. Due to treadmill limitations, We only investigated the prediction of flat to upward sloped terrain. Future work could generalize this by predicting arbitrary slope in the 2D slope configuration space across different types of terrain. Another possible direction would be to create a deep neural network capable of extracting gait information and applying learned features to different applications through transfer learning [9]. Through this scheme, one model could be applied to various topics in motion analysis, including slope prediction and activity recognition.

References

[1] A. Medicine, D. Swain, and C. Brawner, *ACSM's Resource Manual for Guidelines for Exercise Testing and Prescription*, ser. ASCMS Resource Manual for Guidlies for Exercise Testing and Prescription. Wolters Kluwer Health/Lippincott Williams & Wilkins, 2012. [Online]. Available: https://books.google.com/books?id=HZKFm0VrmhYC

[2] M. Hatano and Y. Kitahara, “Research on rescue robots climbing up stairs based on the estimation of slope angles with lrf,” in *2016 IEEE 14th International Workshop on Advanced Motion Control (AMC)*, 2016, pp. 109–114.

[3] M. Islam and E. T. Hsiao-Wecksler, “Detection of gait modes using an artificial neural network during walking with a powered ankle-foot orthosis,” *Journal of Biophysics*, vol. 2016, p. 7984157, Dec 2016. [Online]. Available: https://doi.org/10.1155/2016/7984157

[4] L. Chun-lin, “A tutorial of the wavelet transform,” 2010.

[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” *CoRR*, vol. abs/1409.4842, 2014. [Online]. Available: http://arxiv.org/abs/1409.4842

[7] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2017.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A simple way to prevent neural networks from overfitting,” *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[9] C. Tan, F. Sun, T. Kong, W. Zhang, C. Yang, and C. Liu, “A survey on deep transfer learning,” *CoRR*, vol. abs/1808.01974, 2018. [Online]. Available: http://arxiv.org/abs/1808.01974