# Abstract

# Introduction

### Gesture recognition interfaces

Gestures are a versatile and intuitive method human use to interact. It is also being increasingly looked towards as the next step in human machine interfaces due to its ability to provide more types of input, intuitive method and that it does not necessitate physical contact to function. These can be seen in development of gesture interfaces from vehicles such as Jaguar’s XF Sportbrake and Mercedes’ S-Class to consumer products such as Microsoft’s Kinect and LEAP Motion, Virtual Reality devices and remote triggering of drones. Gestures have been shown to be less disruptive as an interface and are expected to be used significantly as an interface within the next 5 years. Some of the limiting factors to its current usage are reliability and power consumption. These can be seen in the significant reduction in a significant drop in user satisfaction due to increased effort when the system has reliability below a threshold (70% in ). It should also be noted that resolution is associated with reliability due to the presence of movement in images. Power consumption has limited use in untethered and handheld applications. These can be seen in examples such as Google’s project Soli where there was a tradeoff in power consumption and resolution – impacting reliability, or Google glass, whose image sensing and computation consumed roughly 50% of the power budget.

### Report Structure

# Background

### Gesture recognition interface hardware

Some hardware used for gesture recognition include pixel cameras, radar (Soli) or wireless reflections (AllSee), structured light cameras (Kinect), binary gradient cameras and neuromorphic vision systems. This paper will focus on the neuromorphic vision system approach due to be having growing research and accessibility to resources.

### Neuromorphic Vision System (NVS)

Neuromorphic Vision Systems, also known as dynamic vision sensors (DVS) or event-based vision are promising as a solution to reliability and power consumption. These devices employ hardware inspired by biological retinas and only produce data when an event occurs. Specifically, it responds to a spike. A spike occurs when there is a change in logarithmic brightness at a pixel above a threshold. The camera responds by providing the pixel position, timestamp and a polarity of the change. Each pixel acts independently and asynchronously. This allows microsecond resolution and latency, milliwatt power consumption and redundancy removal. Specifically, removal of static objects such as the background.

Power consumption reduction can be attributed to static pixels producing no output and the lack of analogue to digital converters (ADCs) in DVS. The ADC is a result of brightness levels that the NVS lacks, instead relying on a polarity only. Latency reduction and time resolution are a result of pixels being independent and firing when they detect changes rather than waiting for the frame in frame-based methods (pixel cameras, binary gradient cameras and structured light cameras).

NVS has been shown to be highly successful in classification applications. IBM research’s convolutional neural network (CNN) gesture recognition system that achieved 105ms gesture detection latency, 96.5% accuracy and 200mW power consumption for categorizing 11 hand gestures, 95% accuracy on Jester Dataset using Temporal Relational Reasoning by MIT. In addition, MIT also showed NVS could be used for classifying manipulative actions. Localization has also been implemented using NVS by ETH Zurich. Classification of gestures based on hand tracking with 96.96% recognition rate was demonstrated in addition to implementation of a gesture command mode that controlled a cursor by tracking the user’s hand. NVS has also been demonstrated to understand 3D and texture information with multiple implementations of reconstruction of environments both with stereo vision and without. This understanding of depth could be useful to understand gestures due to the need to differentiate between orientation of a hand and understanding which fingers are in front or behind. Due to the event-based format of DVS data, NVS often has different architectures and methods.

### Convolutional Neural Network (CNN)

One of the architectures used in gesture recognition is the convolutional neural network. This is a multilayer feedforward network composed of convolution layers, pooling layers and fully connected layers.

Convolution layers are where the neurons in a layer are calculated by convolving sections of the previous layer with filters (made of weights that will be learned) of a smaller size than the original image. These convolutions occur in the two-dimensional space that makes up an image and the sections are chosen in a systematic way. The window which the filter convolves is shifted along a set number of pixels (stride) and the original image is often padded with zeros to allow extraction at the edges. An activation function is then applied to the result of the convolution to form a convolution layer in the network. The activation function creates non-linearity and allowing the weights to learn in a meaningful manner, causing the result to become a neuron. The activation function chosen often is the Linear rectifier or ReLU function. This function sets any negative value to zero and is often used due to being easy to compute, not saturating the output (gradient tending to zero for extreme values) and causes faster convergence toward optimal weights. Temporal convolutions are also used to correlate features in time.

Pooling layers are layers that reduce the size of the representation, speeding up computation and making feature detection more robust. Pooling layers work by applying a filter where the value of the neuron is calculated as a function of the pixel values covered by its window. Max pooling is often used. The function for max pooling simply returns the maximum value within its window. This function is computationally cheap and allows checking if a feature was detected within the window.

Fully connected layers are where each neuron in a layer is connected to every neuron in the previous layer and are used to form the output. This is because the weights adapt to learn how the features detected in previous layers relate to form the output.

CNN have proven invaluable in image recognition applications and are the predominant architecture used for gesture recognition tasks as well. This is because they can learn complex non-linear functions that are difficult to define using rules.

### Other methods and architectures

In addition to CNN, other methods and architectures have also had success in using gesture recognition, with methods often used in conjunction to each other as stages. These include Vergence, hidden Markov models (HMM), leaky integrate-and-fire neurons (LIF), optical flow, clustering and contour detection.

Hidden Markov models are where there is a hidden state which impacts a visible output and next state with certain probabilities. Hidden Markov models employs Bayesian probability to predict the hidden states given only the output. The model aims to maximize the probability that the output sequence is observed. It does this by calculating the probability of each state, including dependence on previous states, and the probability of each output corresponding to these. It then chooses the state that maximizes the likelihood of the output occurring as the state predicted to have occurred. This has seen success rates above 99% recognition of gestures. The Baum-Welch algorithm is the method used to find these probabilities.

Leaky integrate-and-fire neurons are neurons connect that keep track of an internal state (its membrane potential), adding to it when it receives input events such as spikes and gradually decreasing it otherwise. If the potential exceeds a certain level, it will fire a spike event and reduce its membrane potential. By connecting these neurons to an area of pixels in NVS, this method allows spatiotemporal correlation since movement in an area will cause many events to be received in a short time and the LIF neuron to fire. These therefore act like spatially then temporally convolved layers in a CNN.

# Project Objectives

Much of the literature and applications referenced above are from groups of experienced researchers with larger time frames, resources and with the backing of companies. This makes the possibility of being able to recreate such applications out of reach in this project but suggests some targets.

The performance metric to be used is the Top1 accuracy of recognition of gestures. This means that only the top prediction will be counted. The network will be given input data from a clip with one gesture and the prediction will be compared to the ground truth for the clip. Accuracy is given by the equation below.

This metric was chosen as it is the predominant metric used for gesture recognition tasks and this would allow for effective comparison. Since the datasets have the same number of training examples in each, training and testing data will not be skewed, and the accuracy metric is acceptable.

The target objective will be to achieve 90% accuracy in classification of gestures with initial networks achieving above 70%.

The network classification capability will be compared against the benchmark of the 20BN-JESTER dataset translated into the DVS domain using pix2NVS. 20BN-JESTER has leading accuracy of 96.77% in the pixel domain. Classification capability will also be compared against current DVS classification using IBM’s DVS128 gesture dataset that has classification of 96.49%.

The project will also attempt to perform hand localization using the EgoHands dataset converted into the DVS domain. This is predicted to be unlikely to succeed but an attempt will be made.

The architectures used will be CNN, HMM, Trinary weight networks and deep networks. CNN was chosen as it is a versatile architecture for video recognition, being able to correlate both spatially and temporally using filters. It is also the architecture with which most of the other methods were based upon. HMMs have seen wide success in gesture recognition and is the predominant choice for gesture recognition. Trinary weights were chosen due to the event-based processing hardware, where a comparison of performance at lower precision scales would be valuable for future use. Finally, a deep network based on these approaches would be trained to try to improve performance.

# Current progress

In order to better understand the concepts, the coursera courses on machine learning and deep network specialization was completed. These can be seen at the github link.

# Problem Statement

# Proposed Solution

### Performance Metric

### Benchmark

### Architecture

# Timetable

# Conclusion