EEE 451 Comprehensive Technical Manual

An Exhaustive Deep-Dive into the AI Models, Mathematics, and Implementations of the Smart Attendance System

Topics Covered:

Computer Vision Fundamentals, Convolutional Neural Networks, YuNet Anchor-free Detection, SFace Hypersphere Loss, Cosine Similarity Analytics, Liveness Heuristics, and Full Codebase Deconstruction.

Part I: Theoretical Foundations

To understand the AI utilized in this project, one must first master the fundamental concepts of machine learning and computer vision. The following chapters provide the historical and theoretical background necessary to comprehend YuNet and SFace.

1.1 Artificial Intelligence & Deep Learning

In machine learning, deep learning (DL) focuses on utilizing multilayered neural networks to perform tasks such as classification, regression, and representation learning. The field takes inspiration from biological neuroscience and revolves around stacking artificial neurons into layers and "training" them to process data. The adjective "deep" refers to the use of multiple layers (ranging from three to several hundred or thousands) in the network. Methods used can be supervised, semi-supervised or unsupervised. Some common deep learning network architectures include fully connected networks, deep belief networks, recurrent neural networks, convolutional neural networks, generative adversarial networks, transformers, and neural radiance fields. These architectures have been applied to fields including computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection and board game programs, where they have produced results comparable to and in some cases surpassing human expert performance. Early forms of neural networks were inspired by information processing and distributed communication nodes in biological systems, particularly the human brain. However, current neural networks do not intend to model the brain function of organisms, and are generally seen as low-quality models for that purpose. == Overview == Most modern deep learning models are based on multi-layered neural networks such as convolutional neural networks and transformers, although they can also include propositional formulas or latent variables organized layer-wise in deep generative models such as the nodes in deep belief networks and deep Boltzmann machines. Fundamentally, deep learning refers to a class of machine learning algorithms in which a hierarchy of layers is used to transform input data into a progressively more abstract and composite representation. For example, in an image recognition model, the raw input may be an image (represented as a tensor of pixels). The first representational layer may attempt to identify basic shapes such as lines and circles, the second layer may compose and encode arrangements of edges, the third layer may encode a nose and eyes, and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place at which level on its own. Prior to deep learning, machine learning techniques often involved hand-crafted feature engineering to transform the data into a more suitable representation for a classification algorithm to operate on. In the deep learning approach, features are not hand-crafted and the model discovers useful feature representations from the data automatically. This does not eliminate the need for hand-tuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction. The word "deep" in "deep learning" refers to the number of layers through which the data is transformed. More precisely, deep learning systems have a substantial credit assignment path (CAP) depth. The CAP is the chain of transformations from input to output. CAPs describe potentially causal connections between input and output. For a feedforward neural network, the depth of the CAPs is that of the network and is the number of hidden layers plus one (as the output

layer is also parameterized). For recurrent neural networks, in which a signal may propagate through a layer more than once, the CAP depth is potentially unlimited. No universally agreed-upon threshold of depth divides shallow learning from deep learning, but most researchers agree that deep learning involves CAP depth higher than two. CAP of depth two has been shown to be a universal approximator in the sense that it can emulate any function. Beyond that, more layers do not add to the function approximator ability of the network. Deep models (CAP > two) are able to extract better features than shallow models and hence, extra layers help in learning the features effectively. Deep learning architectures can be constructed with a greedy layer-by-layer method. Deep learning helps to disentangle these abstractions and pick out which features improve performance. Deep learning algorithms can be applied to unsupervised learning tasks. This is an important benefit because unlabeled data is more abundant than the labeled data. Examples of deep structures that can be trained in an unsupervised manner are deep belief networks. The term deep learning was introduced to the machine learning community by Rina Dechter in 1986, and to artificial neural networks by Igor Aizenberg and colleagues in 2000, in the context of Boolean threshold neurons. Although the history of its appearance is apparently more complicated. == Interpretations == Deep neural networks are generally interpreted in terms of the universal approximation theorem or probabilistic inference. The classic universal approximation theorem concerns the capacity of feedforward neural networks with a single hidden layer of finite size to approximate continuous functions. In 1989, the first proof was published by George Cybenko for sigmoid activation functions and was generalised to feed-forward multi-layer architectures in 1991 by Kurt Hornik. Recent work also showed that universal approximation also holds for non-bounded activation functions such as Kunihiko Fukushima's rectified linear unit. The universal approximation theorem for deep neural networks concerns the capacity of networks with bounded width but the depth is allowed to grow. Lu et al. proved that if the width of a deep neural network with ReLU activation is strictly larger than the input dimension, then the network can approximate any Lebesgue integrable function; if the width is smaller or equal to the input dimension, then a deep neural network is not a universal approximator. The probabilistic interpretation derives from the field of machine learning. It features inference, as well as the optimization concepts of training and testing, related to fitting and generalization, respectively. More specifically, the probabilistic interpretation considers the activation nonlinearity as a cumulative distribution function. The probabilistic interpretation led to the introduction of dropout as regularizer in neural networks. The probabilistic interpretation was introduced by researchers including Hopfield, Widrow and Narendra and popularized in surveys such as the one by Bishop. == History == === Before 1980 === There are two types of artificial neural network (ANN): feedforward neural network (FNN) or multilayer perceptron (MLP) and recurrent neural networks (RNN). RNNs have cycles in their connectivity structure, FNNs don't. In the 1920s, Wilhelm Lenz and Ernst Ising created the Ising model which is essentially a non-learning RNN architecture consisting of neuron-like threshold elements. In 1972, Shun'ichi Amari made this architecture adaptive. His learning RNN was republished by John Hopfield in 1982. Other early recurrent neural networks were published by Kaoru Nakano in 1971. Already in 1948, Alan Turing produced work on "Intelligent Machinery" that was not published in his lifetime, containing "ideas related to artificial evolution and learning RNNs". Frank Rosenblatt (1958) proposed the perceptron, an MLP with 3 layers: an input layer, a hidden layer with randomized weights that did not learn, and an output layer. He later published a 1962 book that also introduced variants and computer experiments, including a version with four-layer perceptrons "with adaptive preterminal networks" where the last two layers have learned weights (here he

credits H. D. Block and B. W. Knight). The book cites an earlier network by R. D. Joseph (1960) "functionally equivalent to a variation of" this four-layer system (the book mentions Joseph over 30 times). Should Joseph therefore be considered the originator of proper adaptive multilayer perceptrons with learning hidden units? Unfortunately, the learning algorithm was not a functional one, and fell into oblivion. The first working deep learning algorithm was the Group method of data handling, a method to train arbitrarily deep neural networks, published by Alexey Ivakhnenko and Lapa in 1965. They regarded it as a form of polynomial regression, or a generalization of Rosenblatt's perceptron to handle more complex, nonlinear, and hierarchical relationships. A 1971 paper described a deep network with eight layers trained by this method, which is based on layer by layer training through regression analysis. Superfluous hidden units are pruned using a separate validation set. Since the activation functions of the nodes are Kolmogorov-Gabor polynomials, these were also the first deep networks with multiplicative units or "gates". The first deep learning multilayer perceptron trained by stochastic gradient descent was published in 1967 by Shun'ichi Amari. In computer experiments conducted by Amari's student Saito, a five layer MLP with two modifiable layers learned internal representations to classify non-linearily separable pattern classes. Subsequent developments in hardware and hyperparameter tunings have made end-to-end stochastic gradient descent the currently dominant training technique. In 1969, Kunihiko Fukushima introduced the ReLU (rectified linear unit) activation function. The rectifier has become the most popular activation function for deep learning. Deep learning architectures for convolutional neural networks (CNNs) with convolutional layers and downsampling layers began with the Neocognitron introduced by Kunihiko Fukushima in 1979, though not trained by backpropagation. Backpropagation is an efficient application of the chain rule derived by Gottfried Wilhelm Leibniz in 1673 to networks of differentiable nodes. The terminology "back-propagating errors" was actually introduced in 1962 by Rosenblatt, but he did not know how to implement this, although Henry J. Kelley had a continuous precursor of backpropagation in 1960 in the context of control theory. The modern form of backpropagation was first published in Seppo Linnainmaa's master thesis (1970). G.M. Ostrovski et al. republished it in 1971. Paul Werbos applied backpropagation to neural networks in 1982 (his 1974 PhD thesis, reprinted in a 1994 book, did not yet describe the algorithm). In 1986, David E. Rumelhart et al. popularised backpropagation but did not cite the original work. === 1980s-2000s === The time delay neural network (TDNN) was introduced in 1987 by Alex Waibel to apply CNN to phoneme recognition. It used convolutions, weight sharing, and backpropagation. In 1988, Wei Zhang applied a backpropagation-trained CNN to alphabet recognition. In 1989, Yann LeCun et al. created a CNN called LeNet for recognizing handwritten ZIP codes on mail. Training required 3 days. In 1990, Wei Zhang implemented a CNN on optical computing hardware. In 1991, a CNN was applied to medical image object segmentation and breast cancer detection in mammograms. LeNet-5 (1998), a 7-level CNN by Yann LeCun et al., that classifies digits, was applied by several banks to recognize hand-written numbers on checks digitized in 32x32 pixel images. Recurrent neural networks (RNN) were further developed in the 1980s. Recurrence is used for sequence processing, and when a recurrent network is unrolled, it mathematically resembles a deep feedforward layer. Consequently, they have similar properties and issues, and their developments had mutual influences. In RNN, two early influential works were the Jordan network (1986) and the Elman network (1990), which applied RNN to study problems in cognitive psychology. In the 1980s, backpropagation did not work well for deep learning with long credit assignment paths. To overcome this problem, in 1991, Jürgen Schmidhuber proposed a hierarchy of RNNs pre-trained one level at a time by self-supervised learning where each RNN tries

to predict its own next input, which is the next unexpected input of the RNN below. This "neural history compressor" uses predictive coding to learn internal representations at multiple self-organizing time scales. This can substantially facilitate downstream deep learning. The RNN hierarchy can be collapsed into a single RNN, by distilling a higher level chunker network into a lower level automatizer network. In 1993, a neural history compressor solved a "Very Deep Learning" task that required more than 1000 subsequent layers in an RNN unfolded in time. The "P" in ChatGPT refers to such pre-training. Sepp Hochreiter's diploma thesis (1991) implemented the neural history compressor, and identified and analyzed the vanishing gradient problem. Hochreiter proposed recurrent residual connections to solve the vanishing gradient problem. This led to the long short-term memory (LSTM), published in 1995. LSTM can learn "very deep learning" tasks with long credit assignment paths that require memories of events that happened thousands of discrete time steps before. That LSTM was not yet the modern architecture, which required a "forget gate", introduced in 1999, which became the standard RNN architecture. In 1991, Jürgen Schmidhuber also published adversarial neural networks that contest with each other in the form of a zero-sum game, where one network's gain is the other network's loss. The first network is a generative model that models a probability distribution over output patterns. The second network learns by gradient descent to predict the reactions of the environment to these patterns. This was called "artificial curiosity". In 2014, this principle was used in generative adversarial networks (GANs). During 1985–1995, inspired by statistical mechanics, several architectures and methods were developed by Terry Sejnowski, Peter Dayan, Geoffrey Hinton, etc., including the Boltzmann machine, restricted Boltzmann machine, Helmholtz machine, and the wake-sleep algorithm. These were designed for unsupervised learning of deep generative models. However, those were more computationally expensive compared to backpropagation. Boltzmann machine learning algorithm, published in 1985, was briefly popular before being eclipsed by the backpropagation algorithm in 1986. (p. 112). A 1988 network became state of the art in protein structure prediction, an early application of deep learning to bioinformatics. Both shallow and deep learning (e.g., recurrent nets) of ANNs for speech recognition have been explored for many years. These methods never outperformed non-uniform internal-handcrafting Gaussian mixture model/Hidden Markov model (GMM-HMM) technology based on generative models of speech trained discriminatively. Key difficulties have been analyzed, including gradient diminishing and weak temporal correlation structure in neural predictive models. Additional difficulties were the lack of training data and limited computing power. Most speech recognition researchers moved away from neural nets to pursue generative modeling. An exception was at SRI International in the late 1990s. Funded by the US government's NSA and DARPA, SRI researched in speech and speaker recognition. The speaker recognition team led by Larry Heck reported significant success with deep neural networks in speech processing in the 1998 NIST Speaker Recognition benchmark. It was deployed in the Nuance Verifier, representing the first major industrial application of deep learning. The principle of elevating "raw" features over hand-crafted optimization was first explored successfully in the architecture of deep autoencoder on the "raw" spectrogram or linear filter-bank features in the late 1990s, showing its superiority over the Mel-Cepstral features that contain stages of fixed transformation from spectrograms. The raw features of speech, waveforms, later produced excellent larger-scale results. === 2000s === Neural networks entered a lull, and simpler models that use task-specific handcrafted features such as Gabor filters and support vector machines (SVMs) became the preferred choices in the 1990s and 2000s, because of artificial neural networks' computational cost and a lack of understanding of how the brain wires its biological

networks. In 2003, LSTM became competitive with traditional speech recognizers on certain tasks. In 2006, Alex Graves, Santiago Fernández, Faustino Gomez, and Schmidhuber combined it with connectionist temporal classification (CTC) in stacks of LSTMs. In 2009, it became the first RNN to win a pattern recognition contest, in connected handwriting recognition. In 2006, publications by Geoff Hinton, Ruslan Salakhutdinov, Osindero and Teh deep belief networks were developed for generative modeling. They are trained by training one restricted Boltzmann machine, then freezing it and training another one on top of the first one, and so on, then optionally fine-tuned using supervised backpropagation. They could model high-dimensional probability distributions, such as the distribution of MNIST images, but convergence was slow. The impact of deep learning in industry began in the early 2000s, when CNNs already processed an estimated 10% to 20% of all the checks written in the US, according to Yann LeCun. Industrial applications of deep learning to large-scale speech recognition started around 2010. The 2009 NIPS Workshop on Deep Learning for Speech Recognition was motivated by the limitations of deep generative models of speech, and the possibility that given more capable hardware and large-scale data sets that deep neural nets might become practical. It was believed that pre-training DNNs using generative models of deep belief nets (DBN) would overcome the main difficulties of neural nets. However, it was discovered that replacing pre-training with large amounts of training data for straightforward backpropagation when using DNNs with large, context-dependent output layers produced error rates dramatically lower than then-state-of-the-art Gaussian mixture model (GMM)/Hidden Markov Model (HMM) and also than more-advanced generative model-based systems. The nature of the recognition errors produced by the two types of systems was characteristically different, offering technical insights into how to integrate deep learning into the existing highly efficient, run-time speech decoding system deployed by all major speech recognition systems. Analysis around 2009–2010, contrasting the GMM (and other generative speech models) vs. DNN models, stimulated early industrial investment in deep learning for speech recognition. That analysis was done with comparable performance (less than 1.5% in error rate) between discriminative DNNs and generative models. In 2010, researchers extended deep learning from TIMIT to large vocabulary speech recognition, by adopting large output layers of the DNN based on context-dependent HMM states constructed by decision trees. === Deep learning revolution === The deep learning revolution started around CNN- and GPU-based computer vision. Although CNNs trained by backpropagation had been around for decades and GPU implementations of NNs for years, including CNNs, faster implementations of CNNs on GPUs were needed to progress on computer vision. Later, as deep learning becomes widespread, specialized hardware and algorithm optimizations were developed specifically for deep learning. A key advance for the deep learning revolution was hardware advances, especially GPU. Some early work dated back to 2004. In 2009, Raina, Madhavan, and Andrew Ng reported a 100M deep belief network trained on 30 Nvidia GeForce GTX 280 GPUs, an early demonstration of GPU-based deep learning. They reported up to 70 times faster training. In 2011, a CNN named DanNet by Dan Ciresan, Ueli Meier, Jonathan Masci, Luca Maria Gambardella, and Jürgen Schmidhuber achieved for the first time superhuman performance in a visual pattern recognition contest, outperforming traditional methods by a factor of 3. It then won more contests. They also showed how max-pooling CNNs on GPU improved performance significantly. In 2012, Andrew Ng and Jeff Dean created an FNN that learned to recognize higher-level concepts, such as cats, only from watching unlabeled images taken from YouTube videos. In October 2012, AlexNet by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton won the large-scale ImageNet competition by a significant margin over shallow machine learning methods. Further incremental improvements

included the VGG-16 network by Karen Simonyan and Andrew Zisserman and Google's Inceptionv3. The success in image classification was then extended to the more challenging task of generating descriptions (captions) for images, often as a combination of CNNs and LSTMs. In 2014, the state of the art was training "very deep neural network" with 20 to 30 layers. Stacking too many layers led to a steep reduction in training accuracy, known as the "degradation" problem. In 2015, two techniques were developed to train very deep networks: the highway network was published in May 2015, and the residual neural network (ResNet) in Dec 2015. ResNet behaves like an open-gated Highway Net. Around the same time, deep learning started impacting the field of art. Early examples included Google DeepDream (2015), and neural style transfer (2015), both of which were based on pretrained image classification neural networks, such as VGG-19. Generative adversarial network (GAN) by (Ian Goodfellow et al., 2014) (based on Jürgen Schmidhuber's principle of artificial curiosity) became state of the art in generative modeling during 2014-2018 period. Excellent image quality is achieved by Nvidia's StyleGAN (2018) based on the Progressive GAN by Tero Karras et al. Here the GAN generator is grown from small to large scale in a pyramidal fashion. Image generation by GAN reached popular success, and provoked discussions concerning deepfakes. Diffusion models (2015) eclipsed GANs in generative modeling since then, with systems such as DALL·E 2 (2022) and Stable Diffusion (2022). In 2015, Google's speech recognition improved by 49% by an LSTM-based model, which they made available through Google Voice Search on smartphone. Deep learning is part of state-of-the-art systems in various disciplines, particularly computer vision and automatic speech recognition (ASR). Results on commonly used evaluation sets such as TIMIT (ASR) and MNIST (image classification), as well as a range of large-vocabulary speech recognition tasks have steadily improved. Convolutional neural networks were superseded for ASR by LSTM. but are more successful in computer vision. Yoshua Bengio, Geoffrey Hinton and Yann LeCun were awarded the 2018 Turing Award for "conceptual and engineering breakthroughs that have made deep neural networks a critical component of computing". == Neural networks == Artificial neural networks (ANNs) or connectionist systems are computing systems inspired by the biological neural networks that constitute animal brains. Such systems learn (progressively improve their ability) to do tasks by considering examples, generally without task-specific programming. For example, in image recognition, they might learn to identify images that contain cats by analyzing example images that have been manually labeled as "cat" or "no cat" and using the analytic results to identify cats in other images. They have found most use in applications difficult to express with a traditional computer algorithm using rule-based programming. An ANN is based on a collection of connected units called artificial neurons, (analogous to biological neurons in a biological brain). Each connection (synapse) between neurons can transmit a signal to another neuron. The receiving (postsynaptic) neuron can process the signal(s) and then signal downstream neurons connected to it. Neurons may have state, generally represented by real numbers, typically between 0 and 1. Neurons and synapses may also have a weight that varies as learning proceeds, which can increase or decrease the strength of the signal that it sends downstream. Typically, neurons are organized in layers. Different layers may perform different kinds of transformations on their inputs. Signals travel from the first (input), to the last (output) layer, possibly after traversing the layers multiple times. The original goal of the neural network approach was to solve problems in the same way that a human brain would. Over time, attention focused on matching specific mental abilities, leading to deviations from biology such as backpropagation, or passing information in the reverse direction and adjusting the network to reflect that information. Neural networks have been used on a variety of tasks, including computer vision, speech

recognition, machine translation, social network filtering, playing board and video games and medical diagnosis. As of 2017, neural networks typically have a few thousand to a few million units and millions of connections. Despite this number being several order of magnitude less than the number of neurons on a human brain, these networks can perform many tasks at a level beyond that of humans (e.g., recognizing faces, or playing "Go"). === Deep neural networks === A deep neural network (DNN) is an artificial neural network with multiple layers between the input and output layers. There are different types of neural networks but they always consist of the same components: neurons, synapses, weights, biases, and functions. These components as a whole function in a way that mimics functions of the human brain, and can be trained like any other ML algorithm. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed. The user can review the results and select which probabilities the network should display (above a certain threshold, etc.) and return the proposed label. Each mathematical manipulation as such is considered a layer, and complex DNN have many layers, hence the name "deep" networks. DNNs can model complex non-linear relationships. DNN architectures generate compositional models where the object is expressed as a layered composition of primitives. The extra layers enable composition of features from lower layers, potentially modeling complex data with fewer units than a similarly performing shallow network. For instance, it was proved that sparse multivariate polynomials are exponentially easier to approximate with DNNs than with shallow networks. Deep architectures include many variants of a few basic approaches. Each architecture has found success in specific domains. It is not always possible to compare the performance of multiple architectures, unless they have been evaluated on the same data sets. DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or "weights", to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network did not accurately recognize a particular pattern, an algorithm would adjust the weights. That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data. Recurrent neural networks, in which data can flow in any direction, are used for applications such as language modeling. Long short-term memory is particularly effective for this use. Convolutional neural networks (CNNs) are used in computer vision. CNNs also have been applied to acoustic modeling for automatic speech recognition (ASR). ==== Challenges ==== As with ANNs, many issues can arise with naively trained DNNs. Two common issues are overfitting and computation time. DNNs are prone to overfitting because of the added layers of abstraction, which allow them to model rare dependencies in the training data. Regularization methods such as Ivakhnenko's unit pruning or weight decay (■ 2 {\displaystyle \ell _{2}} -regularization) or sparsity (■ 1 {\displaystyle \ell _{1}} -regularization) can be applied during training to combat overfitting. Alternatively dropout regularization randomly omits units from the hidden layers during training. This helps to exclude rare dependencies. Another interesting recent development is research into models of just enough complexity through an estimation of the intrinsic complexity of the task being modelled. This approach has been successfully applied for multivariate time series prediction tasks such as traffic prediction. Finally, data can be augmented via methods such as cropping and rotating such that smaller training sets can be increased in size to reduce the chances of overfitting. DNNs must consider many training parameters, such as the size (number of layers and number of units per layer), the learning rate, and initial weights. Sweeping through the parameter space for optimal parameters may not be feasible due to the cost

in time and computational resources. Various tricks, such as batching (computing the gradient on several training examples at once rather than individual examples) speed up computation. Large processing capabilities of many-core architectures (such as GPUs or the Intel Xeon Phi) have produced significant speedups in training, because of the suitability of such processing architectures for the matrix and vector computations. Alternatively, engineers may look for other types of neural networks with more straightforward and convergent training algorithms. CMAC (cerebellar model articulation controller) is one such kind of neural network. It doesn't require learning rates or randomized initial weights. The training process can be guaranteed to converge in one step with a new batch of data, and the computational complexity of the training algorithm is linear with respect to the number of neurons involved. == Hardware == Since the 2010s, advances in both machine learning algorithms and computer hardware have led to more efficient methods for training deep neural networks that contain many layers of non-linear hidden units and a very large output layer. By 2019, graphics processing units (GPUs), often with AI-specific enhancements, had displaced CPUs as the dominant method for training large-scale commercial cloud AI . OpenAI estimated the hardware computation used in the largest deep learning projects from AlexNet (2012) to AlphaZero (2017) and found a 300,000-fold increase in the amount of computation required, with a doubling-time trendline of 3.4 months. Special electronic circuits called deep learning processors were designed to speed up deep learning algorithms. Deep learning processors include neural processing units (NPUs) in Huawei cellphones and cloud computing servers such as tensor processing units (TPU) in the Google Cloud Platform. Cerebras Systems has also built a dedicated system to handle large deep learning models, the CS-2, based on the largest processor in the industry, the second-generation Wafer Scale Engine (WSE-2). Atomically thin semiconductors are considered promising for energy-efficient deep learning hardware where the same basic device structure is used for both logic operations and data storage. In 2020, Marega et al. published experiments with a large-area active channel material for developing logic-in-memory devices and circuits based on floating-gate field-effect transistors (FGFETs). In 2021, J. Feldmann et al. proposed an integrated photonic hardware accelerator for parallel convolutional processing. The authors identify two key advantages of integrated photonics over its electronic counterparts: (1) massively parallel data transfer through wavelength division multiplexing in conjunction with frequency combs, and (2) extremely high data modulation speeds. Their system can execute trillions of multiply-accumulate operations per second, indicating the potential of integrated photonics in data-heavy AI applications. == Applications == === Automatic speech recognition === Large-scale automatic speech recognition is the first and most convincing successful case of deep learning. LSTM RNNs can learn "Very Deep Learning" tasks that involve multi-second intervals containing speech events separated by thousands of discrete time steps, where one time step corresponds to about 10 ms. LSTM with forget gates is competitive with traditional speech recognizers on certain tasks. The initial success in speech recognition was based on small-scale recognition tasks based on TIMIT. The data set contains 630 speakers from eight major dialects of American English, where each speaker reads 10 sentences. Its small size lets many configurations be tried. More importantly, the TIMIT task concerns phone-sequence recognition, which, unlike word-sequence recognition, allows weak phone bigram language models. This lets the strength of the acoustic modeling aspects of speech recognition be more easily analyzed. The error rates listed below, including these early results and measured as percent phone error rates (PER), have been summarized since 1991. The debut of DNNs for speaker recognition in the late 1990s and speech recognition around 2009-2011 and of LSTM around 2003–2007, accelerated progress in eight

major areas: Scale-up/out and accelerated DNN training and decoding Sequence discriminative training Feature processing by deep models with solid understanding of the underlying mechanisms Adaptation of DNNs and related deep models Multi-task and transfer learning by DNNs and related deep models CNNs and how to design them to best exploit domain knowledge of speech RNN and its rich LSTM variants Other types of deep models including tensor-based models and integrated deep generative/discriminative models. More recent speech recognition models use Transformers or Temporal Convolution Networks with significant success and widespread applications. All major commercial speech recognition systems (e.g., Microsoft Cortana, Xbox, Skype Translator, Amazon Alexa, Google Now, Apple Siri, Baidu and iFlyTek voice search, and a range of Nuance speech products, etc.) are based on deep learning. === Image recognition === A common evaluation set for image classification is the MNIST database data set. MNIST is composed of handwritten digits and includes 60,000 training examples and 10,000 test examples. As with TIMIT, its small size lets users test multiple configurations. A comprehensive list of results on this set is available. Deep learning-based image recognition has become "superhuman", producing more accurate results than human contestants. This first occurred in 2011 in recognition of traffic signs, and in 2014, with recognition of human faces. Deep learning-trained vehicles now interpret 360° camera views. Another example is Facial Dysmorphology Novel Analysis (FDNA) used to analyze cases of human malformation connected to a large database of genetic syndromes. === Visual art processing === Closely related to the progress that has been made in image recognition is the increasing application of deep learning techniques to various visual art tasks. DNNs have proven themselves capable, for example, of identifying the style period of a given painting Neural Style Transfer – capturing the style of a given artwork and applying it in a visually pleasing manner to an arbitrary photograph or video generating striking imagery based on random visual input fields. === Natural language processing === Neural networks have been used for implementing language models since the early 2000s. LSTM helped to improve machine translation and language modeling. Other key techniques in this field are negative sampling and word embedding. Word embedding, such as word2vec, can be thought of as a representational layer in a deep learning architecture that transforms an atomic word into a positional representation of the word relative to other words in the dataset; the position is represented as a point in a vector space. Using word embedding as an RNN input layer allows the network to parse sentences and phrases using an effective compositional vector grammar. A compositional vector grammar can be thought of as probabilistic context free grammar (PCFG) implemented by an RNN. Recursive auto-encoders built atop word embeddings can assess sentence similarity and detect paraphrasing. Deep neural architectures provide the best results for constituency parsing, sentiment analysis, information retrieval, spoken language understanding, machine translation, contextual entity linking, writing style recognition, named-entity recognition (token classification), text classification, and others. Recent developments generalize word embedding to sentence embedding. Google Translate (GT) uses a large end-to-end long short-term memory (LSTM) network. Google Neural Machine Translation (GNMT) uses an example-based machine translation method in which the system "learns from millions of examples". It translates "whole sentences at a time, rather than pieces". Google Translate supports over one hundred languages. The network encodes the "semantics of the sentence rather than simply memorizing phrase-to-phrase translations". GT uses English as an intermediate between most language pairs. === Drug discovery and toxicology === A large percentage of candidate drugs fail to win regulatory approval. These failures are caused by insufficient efficacy (on-target effect), undesired interactions (off-target effects), or unanticipated toxic effects. Research has explored use

of deep learning to predict the biomolecular targets, off-targets, and toxic effects of environmental chemicals in nutrients, household products and drugs. AtomNet is a deep learning system for structure-based rational drug design. AtomNet was used to predict novel candidate biomolecules for disease targets such as the Ebola virus and multiple sclerosis. In 2017 graph neural networks were used for the first time to predict various properties of molecules in a large toxicology data set. In 2019, generative neural networks were used to produce molecules that were validated experimentally all the way into mice. === Recommendation systems === Recommendation systems have used deep learning to extract meaningful features for a latent factor model for content-based music and journal recommendations. Multi-view deep learning has been applied for learning user preferences from multiple domains. The model uses a hybrid collaborative and content-based approach and enhances recommendations in multiple tasks. === Bioinformatics === An autoencoder ANN was used in bioinformatics, to predict gene ontology annotations and gene-function relationships. In medical informatics, deep learning was used to predict sleep quality based on data from wearables and predictions of health complications from electronic health record data. Deep neural networks have shown unparalleled performance in predicting protein structure, according to the sequence of the amino acids that make it up. In 2020, AlphaFold, a deep-learning based system, achieved a level of accuracy significantly higher than all previous computational methods. === Deep Neural Network Estimations === Deep neural networks can be used to estimate the entropy of a stochastic process through an arrangement called a Neural Joint Entropy Estimator (NJEE). Such an estimation provides insights on the effects of input random variables on an independent random variable. Practically, the DNN is trained as a classifier that maps an input vector or matrix X to an output probability distribution over the possible classes of random variable Y, given input X. For example, in image classification tasks, the NJEE maps a vector of pixels' color values to probabilities over possible image classes. In practice, the probability distribution of Y is obtained by a Softmax layer with number of nodes that is equal to the alphabet size of Y. NJEE uses continuously differentiable activation functions, such that the conditions for the universal approximation theorem holds. It is shown that this method provides a strongly consistent estimator and outperforms other methods in cases of large alphabet sizes. === Medical image analysis === Deep learning has been shown to produce competitive results in medical applications such as cancer cell classification, lesion detection, organ segmentation and image enhancement. Modern deep learning tools demonstrate the high accuracy of detecting various diseases and the helpfulness of their use by specialists to improve the diagnosis efficiency. === Mobile advertising === Finding the appropriate mobile audience for mobile advertising is always challenging, since many data points must be considered and analyzed before a target segment can be created and used in ad serving by any ad server. Deep learning has been used to interpret large, many-dimensioned advertising datasets. Many data points are collected during the request/serve/click internet advertising cycle. This information can form the basis of machine learning to improve ad selection. === Image restoration === Deep learning has been successfully applied to inverse problems such as denoising, super-resolution, inpainting, and film colorization. These applications include learning methods such as "Shrinkage Fields for Effective Image Restoration" which trains on an image dataset, and Deep Image Prior, which trains on the image that needs restoration. === Financial fraud detection === Deep learning is being successfully applied to financial fraud detection, tax evasion detection, and anti-money laundering. === Materials science === In November 2023, researchers at Google DeepMind and Lawrence Berkeley National Laboratory announced that they had developed an AI system known as GNoME.

This system has contributed to materials science by discovering over 2 million new materials within a relatively short timeframe. GNoME employs deep learning techniques to efficiently explore potential material structures, achieving a significant increase in the identification of stable inorganic crystal structures. The system's predictions were validated through autonomous robotic experiments, demonstrating a noteworthy success rate of 71%. The data of newly discovered materials is publicly available through the Materials Project database, offering researchers the opportunity to identify materials with desired properties for various applications. This development has implications for the future of scientific discovery and the integration of AI in material science research, potentially expediting material innovation and reducing costs in product development. The use of AI and deep learning suggests the possibility of minimizing or eliminating manual lab experiments and allowing scientists to focus more on the design and analysis of unique compounds. === Military === The United States Department of Defense applied deep learning to train robots in new tasks through observation. === Partial differential equations === Physics informed neural networks have been used to solve partial differential equations in both forward and inverse problems in a data driven manner. One example is the reconstructing fluid flow governed by the Navier-Stokes equations. Using physics informed neural networks does not require the often expensive mesh generation that conventional CFD methods rely on. It is evident that geometric and physical constraints have a synergistic effect on neural PDE surrogates, thereby enhancing their efficacy in predicting stable and super long rollouts. === Deep backward stochastic differential equation method === Deep backward stochastic differential equation method is a numerical method that combines deep learning with Backward stochastic differential equation (BSDE). This method is particularly useful for solving high-dimensional problems in financial mathematics. By leveraging the powerful function approximation capabilities of deep neural networks, deep BSDE addresses the computational challenges faced by traditional numerical methods in high-dimensional settings. Specifically, traditional methods like finite difference methods or Monte Carlo simulations often struggle with the curse of dimensionality, where computational cost increases exponentially with the number of dimensions. Deep BSDE methods, however, employ deep neural networks to approximate solutions of high-dimensional partial differential equations (PDEs), effectively reducing the computational burden. In addition, the integration of Physics-informed neural networks (PINNs) into the deep BSDE framework enhances its capability by embedding the underlying physical laws directly into the neural network architecture. This ensures that the solutions not only fit the data but also adhere to the governing stochastic differential equations. PINNs leverage the power of deep learning while respecting the constraints imposed by the physical models, resulting in more accurate and reliable solutions for financial mathematics problems. === Image reconstruction === Image reconstruction is the reconstruction of the underlying images from the image-related measurements. Several works showed the better and superior performance of the deep learning methods compared to analytical methods for various applications, e.g., spectral imaging and ultrasound imaging. === Weather prediction === Traditional weather prediction systems solve a very complex system of partial differential equations. GraphCast is a deep learning based model, trained on a long history of weather data to predict how weather patterns change over time. It is able to predict weather conditions for up to 10 days globally, at a very detailed level, and in under a minute, with precision similar to state of the art systems. === Epigenetic clock === An epigenetic clock is a biochemical test that can be used to measure age. Galkin et al. used deep neural networks to train an epigenetic aging clock of unprecedented accuracy using >6,000 blood samples. The clock uses information from 1000 CpG

sites and predicts people with certain conditions older than healthy controls: IBD, frontotemporal dementia, ovarian cancer, obesity. The aging clock was planned to be released for public use in 2021 by an Insilico Medicine spinoff company Deep Longevity. == Relation to human cognitive and brain development == Deep learning is closely related to a class of theories of brain development (specifically, neocortical development) proposed by cognitive neuroscientists in the early 1990s. These developmental theories were instantiated in computational models, making them predecessors of deep learning systems. These developmental models share the property that various proposed learning dynamics in the brain (e.g., a wave of nerve growth factor) support the self-organization somewhat analogous to the neural networks utilized in deep learning models. Like the neocortex, neural networks employ a hierarchy of layered filters in which each layer considers information from a prior layer (or the operating environment), and then passes its output (and possibly the original input), to other layers. This process yields a self-organizing stack of transducers, well-tuned to their operating environment. A 1995 description stated, "...the infant's brain seems to organize itself under the influence of waves of so-called trophic-factors ... different regions of the brain become connected sequentially, with one layer of tissue maturing before another and so on until the whole brain is mature". A variety of approaches have been used to investigate the plausibility of deep learning models from a neurobiological perspective. On the one hand, several variants of the backpropagation algorithm have been proposed in order to increase its processing realism. Other researchers have argued that unsupervised forms of deep learning, such as those based on hierarchical generative models and deep belief networks, may be closer to biological reality. In this respect, generative neural network models have been related to neurobiological evidence about sampling-based processing in the cerebral cortex. Although a systematic comparison between the human brain organization and the neuronal encoding in deep networks has not yet been established, several analogies have been reported. For example, the computations performed by deep learning units could be similar to those of actual neurons and neural populations. Similarly, the representations developed by deep learning models are similar to those measured in the primate visual system both at the single-unit and at the population levels. == Commercial activity == Facebook's AI lab performs tasks such as automatically tagging uploaded pictures with the names of the people in them. Google's DeepMind Technologies developed a system capable of learning how to play Atari video games using only pixels as data input. In 2015 they demonstrated their AlphaGo system, which learned the game of Go well enough to beat a professional Go player. Google Translate uses a neural network to translate between more than 100 languages. In 2017, Covariant.ai was launched, which focuses on integrating deep learning into factories. As of 2008, researchers at The University of Texas at Austin (UT) developed a machine learning framework called Training an Agent Manually via Evaluative Reinforcement, or TAMER, which proposed new methods for robots or computer programs to learn how to perform tasks by interacting with a human instructor. First developed as TAMER, a new algorithm called Deep TAMER was later introduced in 2018 during a collaboration between U.S. Army Research Laboratory (ARL) and UT researchers. Deep TAMER used deep learning to provide a robot with the ability to learn new tasks through observation. Using Deep TAMER, a robot learned a task with a human trainer, watching video streams or observing a human perform a task in-person. The robot later practiced the task with the help of some coaching from the trainer, who provided feedback such as "good job" and "bad job". == Criticism and comment == Deep learning has attracted both criticism and comment, in some cases from outside the field of computer science. === Theory === A main criticism concerns the lack of theory surrounding some methods. Learning in the most

common deep architectures is implemented using well-understood gradient descent. However, the theory surrounding other algorithms, such as contrastive divergence is less clear. (e.g., Does it converge? If so, how fast? What is it approximating?) Deep learning methods are often looked at as a black box, with most confirmations done empirically, rather than theoretically. In further reference to the idea that artistic sensitivity might be inherent in relatively low levels of the cognitive hierarchy, a published series of graphic representations of the internal states of deep (20-30 layers) neural networks attempting to discern within essentially random data the images on which they were trained demonstrate a visual appeal: the original research notice received well over 1,000 comments, and was the subject of what was for a time the most frequently accessed article on The Guardian's website. With the support of Innovation Diffusion Theory (IDT), a study analyzed the diffusion of Deep Learning in BRICS and OECD countries using data from Google Trends. === Errors === Some deep learning architectures display problematic behaviors, such as confidently classifying unrecognizable images as belonging to a familiar category of ordinary images (2014) and misclassifying minuscule perturbations of correctly classified images (2013). Goertzel hypothesized that these behaviors are due to limitations in their internal representations and that these limitations would inhibit integration into heterogeneous multi-component artificial general intelligence (AGI) architectures. These issues may possibly be addressed by deep learning architectures that internally form states homologous to image-grammar decompositions of observed entities and events. Learning a grammar (visual or linguistic) from training data would be equivalent to restricting the system to commonsense reasoning that operates on concepts in terms of grammatical production rules and is a basic goal of both human language acquisition and artificial intelligence (AI). === Cyber threat === As deep learning moves from the lab into the world, research and experience show that artificial neural networks are vulnerable to hacks and deception. By identifying patterns that these systems use to function, attackers can modify inputs to ANNs in such a way that the ANN finds a match that human observers would not recognize. For example, an attacker can make subtle changes to an image such that the ANN finds a match even though the image looks to a human nothing like the search target. Such manipulation is termed an "adversarial attack". In 2016 researchers used one ANN to doctor images in trial and error fashion, identify another's focal points, and thereby generate images that deceived it. The modified images looked no different to human eyes. Another group showed that printouts of doctored images then photographed successfully tricked an image classification system. One defense is reverse image search, in which a possible fake image is submitted to a site such as TinEye that can then find other instances of it. A refinement is to search using only parts of the image, to identify images from which that piece may have been taken. Another group showed that certain psychedelic spectacles could fool a facial recognition system into thinking ordinary people were celebrities, potentially allowing one person to impersonate another. In 2017 researchers added stickers to stop signs and caused an ANN to misclassify them. ANNs can however be further trained to detect attempts at deception, potentially leading attackers and defenders into an arms race similar to the kind that already defines the malware defense industry. ANNs have been trained to defeat ANN-based anti-malware software by repeatedly attacking a defense with malware that was continually altered by a genetic algorithm until it tricked the anti-malware while retaining its ability to damage the target. In 2016, another group demonstrated that certain sounds could make the Google Now voice command system open a particular web address, and hypothesized that this could "serve as a stepping stone for further attacks (e.g., opening a web page hosting drive-by malware)". In "data poisoning", false data is continually smuggled into a machine learning system's training set to

prevent it from achieving mastery. === Data collection ethics === The deep learning systems that are trained using supervised learning often rely on data that is created or annotated by humans, or both. It has been argued that not only low-paid clickwork (such as on Amazon Mechanical Turk) is regularly deployed for this purpose, but also implicit forms of human microwork that are often not recognized as such. The philosopher Rainer Mühlhoff distinguishes five types of "machinic capture" of human microwork to generate training data: (1) gamification (the embedding of annotation or computation tasks in the flow of a game), (2) "trapping and tracking" (e.g. CAPTCHAs for image recognition or click-tracking on Google search results pages), (3) exploitation of social motivations (e.g. tagging faces on Facebook to obtain labeled facial images), (4) information mining (e.g. by leveraging quantified-self devices such as activity trackers) and (5) clickwork. == See also == Applications of artificial intelligence Comparison of deep learning software Compressed sensing Differentiable programming Echo state network List of artificial intelligence projects Liquid state machine List of datasets for machine-learning research Reservoir computing Scale space and deep learning Sparse coding Stochastic parrot Topological deep learning == References == == Further reading ==

1.2 Convolutional Neural Networks (CNNs)

A convolutional neural network (CNN) is a type of feedforward neural network that learns features via filter (or kernel) optimization. This type of deep learning network has been applied to process and make predictions from many different types of data including text, images and audio. CNNs are the de-facto standard in deep learning-based approaches to computer vision and image processing, and have only recently been replaced—in some cases—by newer deep learning architectures such as the transformer. Vanishing gradients and exploding gradients, seen during backpropagation in earlier neural networks, are prevented by the regularization that comes from using shared weights over fewer connections. For example, for each neuron in the fully-connected layer, 10,000 weights would be required for processing an image sized 100 × 100 pixels. However, applying cascaded convolution (or cross-correlation) kernels, only 25 weights for each convolutional layer are required to process 5x5-sized tiles. Higher-layer features are extracted from wider context windows, compared to lower-layer features. Some applications of CNNs include: image and video recognition, recommender systems, image classification, image segmentation, medical image analysis, natural language processing, brain–computer interfaces, and financial time series. CNNs are also known as shift invariant or space invariant artificial neural networks, based on the shared-weight architecture of the convolution kernels or filters that slide along input features and provide translation-equivariant responses known as feature maps. Counter-intuitively, most convolutional neural networks are not invariant to translation, due to the downsampling operation they apply to the input. Feedforward neural networks are usually fully connected networks, that is, each neuron in one layer is connected to all neurons in the next layer. The "full connectivity" of these networks makes them prone to overfitting data. Typical ways of regularization, or preventing overfitting, include: penalizing parameters during training (such as weight decay) or trimming connectivity (skipped connections, dropout, etc.) Robust datasets also increase the probability that CNNs will learn the generalized principles that characterize a given dataset rather than the biases of a poorly-populated set. Convolutional networks were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns to optimize the filters (or kernels) through automated learning, whereas in traditional algorithms these filters are hand-engineered. This simplifies and automates the process, enhancing efficiency and scalability overcoming human-intervention bottlenecks. == Architecture == A convolutional neural network consists of an input layer, hidden layers and an output layer. In a convolutional neural network, the hidden layers include one or more layers that perform convolutions. Typically this includes a layer that performs a dot product of the convolution kernel with the layer's input matrix. This product is usually the Frobenius inner product, and its activation function is commonly ReLU. As the convolution kernel slides along the input matrix for the layer, the convolution operation generates a feature map, which in turn contributes to the input of the next layer. This is followed by other layers such as pooling layers, fully connected layers, and normalization layers. Here it should be noted how close a convolutional neural network is to a matched filter. === Convolutional layers === In a CNN, the input is a tensor with shape: (number of inputs) × (input height) × (input width) × (input channels) After passing through a convolutional layer, the image becomes abstracted to a feature map, also

called an activation map, with shape: (number of inputs) × (feature map height) × (feature map width) × (feature map channels). Convolutional layers convolve the input and pass its result to the next layer. This is similar to the response of a neuron in the visual cortex to a specific stimulus. Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features and classify data, this architecture is generally impractical for larger inputs (e.g., high-resolution images), which would require massive numbers of neurons because each pixel is a relevant input feature. A fully connected layer for an image of size 100 × 100 has 10,000 weights for each neuron in the second layer. Convolution reduces the number of free parameters, allowing the network to be deeper. For example, using a 5 × 5 tiling region, each with the same shared weights, requires only 25 neurons. Using shared weights means there are many fewer parameters, which helps avoid the vanishing gradients and exploding gradients problems seen during backpropagation in earlier neural networks. To speed processing, standard convolutional layers can be replaced by depthwise separable convolutional layers, which are based on a depthwise convolution followed by a pointwise convolution. The depthwise convolution is a spatial convolution applied independently over each channel of the input tensor, while the pointwise convolution is a standard convolution restricted to the use of 1 × 1 {\displaystyle 1\times 1} kernels. === Pooling layers === Convolutional networks may include local and/or global pooling layers along with traditional convolutional layers. Pooling layers reduce the dimensions of data by combining the outputs of neuron clusters at one layer into a single neuron in the next layer. Local pooling combines small clusters, tiling sizes such as 2 × 2 are commonly used. Global pooling acts on all the neurons of the feature map. There are two common types of pooling in popular use: max and average. Max pooling uses the maximum value of each local cluster of neurons in the feature map, while average pooling takes the average value. === Fully connected layers === Fully connected layers connect every neuron in one layer to every neuron in another layer. It is the same as a traditional multilayer perceptron neural network (MLP). Each neuron in the fully connected layer receives input from all the neurons in the previous layer. These inputs are weighted and summed with the corresponding biases, and then passed through an activation function to perform a nonlinear transformation, generating the output. The flattened matrix goes through a fully connected layer to classify the images. === Receptive field === In neural networks, each neuron receives input from some number of locations in the previous layer. In a convolutional layer, each neuron receives input from only a restricted area of the previous layer called the neuron's receptive field. Typically the area is a square (e.g. 5 by 5 neurons). Whereas, in a fully connected layer, the receptive field is the entire previous layer. Thus, in each convolutional layer, each neuron takes input from a larger area in the input than previous layers. This is due to applying the convolution over and over, which takes the value of a pixel into account, as well as its surrounding pixels. When using dilated layers, the number of pixels in the receptive field remains constant, but the field is more sparsely populated as its dimensions grow when combining the effect of several layers. To manipulate the receptive field size as desired, there are some alternatives to the standard convolutional layer. For example, atrous or dilated convolution expands the receptive field size without increasing the number of parameters by interleaving visible and blind regions. Moreover, a single dilated convolutional layer can comprise filters with multiple dilation ratios, thus having a variable receptive field size. === Weights === Each neuron in a neural network computes an output value by applying a specific function to the input values received from the receptive field in the previous layer. The function that is applied to the input values is determined by a vector of weights and a bias (typically real numbers). Learning consists of iteratively adjusting these biases

and weights. The vectors of weights and biases are called filters and represent particular features of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons can share the same filter. This reduces the memory footprint because a single bias and a single vector of weights are used across all receptive fields that share that filter, as opposed to each receptive field having its own bias and vector weighting. === Deconvolutional === A deconvolutional neural network is essentially the reverse of a CNN. It consists of deconvolutional layers and unpooling layers. A deconvolutional layer is the transpose of a convolutional layer. Specifically, a convolutional layer can be written as a multiplication with a matrix, and a deconvolutional layer is multiplication with the transpose of that matrix. An unpooling layer expands the layer. The max-unpooling layer is the simplest, as it simply copies each entry multiple times. For example, a 2-by-2 max-unpooling layer is [x] ■ [x x x x] {\displaystyle [x]\mapsto {\begin{bmatrix}x&x;\\x&x;\end{bmatrix}}} . Deconvolution layers are used in image generators. By default, it creates periodic checkerboard artifact, which can be fixed by upscale-then-convolve. == History == CNN are often compared to the way the brain achieves vision processing in living organisms. === Receptive fields in the visual cortex === Work by Hubel and Wiesel in the 1950s and 1960s showed that cat visual cortices contain neurons that individually respond to small regions of the visual field. Provided the eyes are not moving, the region of visual space within which visual stimuli affect the firing of a single neuron is known as its receptive field. Neighboring cells have similar and overlapping receptive fields. Receptive field size and location varies systematically across the cortex to form a complete map of visual space. The cortex in each hemisphere represents the contralateral visual field. Their 1968 paper identified two basic visual cell types in the brain: simple cells, whose output is maximized by straight edges having particular orientations within their receptive field complex cells, which have larger receptive fields, whose output is insensitive to the exact position of the edges in the field. Hubel and Wiesel also proposed a cascading model of these two types of cells for use in pattern recognition tasks. === Fukushima's analog threshold elements in a vision model === In 1969, Kunihiko Fukushima introduced a multilayer visual feature detection network, inspired by the above-mentioned work of Hubel and Wiesel, in which "All the elements in one layer have the same set of interconnecting coefficients; the arrangement of the elements and their interconnections are all homogeneous over a given layer." This is the essential core of a convolutional network, but the weights were not trained. In the same paper, Fukushima also introduced the ReLU (rectified linear unit) activation function. === Neocognitron, origin of the trainable CNN architecture === The "neocognitron" was introduced by Fukushima in 1980. The neocognitron introduced the two basic types of layers: "S-layer": a shared-weights receptive-field layer, later known as a convolutional layer, which contains units whose receptive fields cover a patch of the previous layer. A shared-weights receptive-field group (a "plane" in neocognitron terminology) is often called a filter, and a layer typically has several such filters. "C-layer": a downsampling layer that contain units whose receptive fields cover patches of previous convolutional layers. Such a unit typically computes a weighted average of the activations of the units in its patch, and applies inhibition (divisive normalization) pooled from a somewhat larger patch and across different filters in a layer, and applies a saturating activation function. The patch weights are nonnegative and are not trainable in the original neocognitron. The downsampling and competitive inhibition help to classify features and objects in visual scenes even when the objects are shifted. Several supervised and unsupervised learning algorithms have been proposed over the decades to train the weights of a neocognitron. Today, however, the CNN architecture is usually trained through backpropagation. Fukushima's ReLU activation function was

not used in his neocognitron since all the weights were nonnegative; lateral inhibition was used instead. The rectifier has become a very popular activation function for CNNs and deep neural networks in general. === Convolution in time === The term "convolution" first appears in neural networks in a paper by Toshiteru Homma, Les Atlas, and Robert Marks II at the first Conference on Neural Information Processing Systems in 1987. Their paper replaced multiplication with convolution in time, inherently providing shift invariance, motivated by and connecting more directly to the signal-processing concept of a filter, and demonstrated it on a speech recognition task. They also pointed out that as a data-trainable system, convolution is essentially equivalent to correlation since reversal of the weights does not affect the final learned function ("For convenience, we denote * as correlation instead of convolution. Note that convolving a(t) with b(t) is equivalent to correlating a(-t) with b(t)."). Modern CNN implementations typically do correlation and call it convolution, for convenience, as they did here. === Time delay neural networks === The time delay neural network (TDNN) was introduced in 1987 by Alex Waibel et al. for phoneme recognition and was an early convolutional network exhibiting shift-invariance. A TDNN is a 1-D convolutional neural net where the convolution is performed along the time axis of the data. It is the first CNN utilizing weight sharing in combination with a training by gradient descent, using backpropagation. Thus, while also using a pyramidal structure as in the neocognitron, it performed a global optimization of the weights instead of a local one. TDNNs are convolutional networks that share weights along the temporal dimension. They allow speech signals to be processed time-invariantly. In 1990 Hampshire and Waibel introduced a variant that performs a two-dimensional convolution. Since these TDNNs operated on spectrograms, the resulting phoneme recognition system was invariant to both time and frequency shifts, as with images processed by a neocognitron. TDNNs improved the performance of far-distance speech recognition. === Image recognition with CNNs trained by gradient descent === Denker et al. (1989) designed a 2-D CNN system to recognize hand-written ZIP Code numbers. However, the lack of an efficient training method to determine the kernel coefficients of the involved convolutions meant that all the coefficients had to be laboriously hand-designed. Following the advances in the training of 1-D CNNs by Waibel et al. (1987), Yann LeCun et al. (1989) used back-propagation to learn the convolution kernel coefficients directly from images of hand-written numbers. Learning was thus fully automatic, performed better than manual coefficient design, and was suited to a broader range of image recognition problems and image types. Wei Zhang et al. (1988) used back-propagation to train the convolution kernels of a CNN for alphabets recognition. The model was called shift-invariant pattern recognition neural network before the name CNN was coined later in the early 1990s. Wei Zhang et al. also applied the same CNN without the last fully connected layer for medical image object segmentation (1991) and breast cancer detection in mammograms (1994). This approach became a foundation of modern computer vision. ==== Max pooling ==== In 1990 Yamaguchi et al. introduced the concept of max pooling, a fixed filtering operation that calculates and propagates the maximum value of a given region. They did so by combining TDNNs with max pooling to realize a speaker-independent isolated word recognition system. In their system they used several TDNNs per word, one for each syllable. The results of each TDNN over the input signal were combined using max pooling and the outputs of the pooling layers were then passed on to networks performing the actual word classification. In a variant of the neocognitron called the cresceptron, instead of using Fukushima's spatial averaging with inhibition and saturation, J. Weng et al. in 1993 used max pooling, where a downsampling unit computes the maximum of the activations of the units in its patch, introducing this method into the vision field. Max pooling is often used in modern CNNs. ==== LeNet-5 ==== LeNet-5, a pioneering

7-level convolutional network by LeCun et al. in 1995, classifies hand-written numbers on checks digitized in 32×32 pixel images. The ability to process higher-resolution images requires larger and more layers of convolutional neural networks, so this technique is constrained by the availability of computing resources. It was superior than other commercial courtesy amount reading systems (as of 1995). The system was integrated in NCR's check reading systems, and fielded in several American banks since June 1996, reading millions of checks per day. === Shift-invariant neural network === A shift-invariant neural network was proposed by Wei Zhang et al. for image character recognition in 1988. It is a modified Neocognitron by keeping only the convolutional interconnections between the image feature layers and the last fully connected layer. The model was trained with back-propagation. The training algorithm was further improved in 1991 to improve its generalization ability. The model architecture was modified by removing the last fully connected layer and applied for medical image segmentation (1991) and automatic detection of breast cancer in mammograms (1994). A different convolution-based design was proposed in 1988 for application to decomposition of one-dimensional electromyography convolved signals via de-convolution. This design was modified in 1989 to other de-convolution-based designs. === GPU implementations === Although CNNs were invented in the 1980s, their breakthrough in the 2000s required fast implementations on graphics processing units (GPUs). In 2004, it was shown by K. S. Oh and K. Jung that standard neural networks can be greatly accelerated on GPUs. Their implementation was 20 times faster than an equivalent implementation on CPU. In 2005, another paper also emphasised the value of GPGPU for machine learning. The first GPU-implementation of a CNN was described in 2006 by K. Chellapilla et al. Their implementation was 4 times faster than an equivalent implementation on CPU. In the same period, GPUs were also used for unsupervised training of deep belief networks. In 2010, Dan Ciresan et al. at IDSIA trained deep feedforward networks on GPUs. In 2011, they extended this to CNNs, accelerating by 60 compared to training CPU. In 2011, the network won an image recognition contest where they achieved superhuman performance for the first time. Then they won more competitions and achieved state of the art on several benchmarks. Subsequently, AlexNet, a similar GPU-based CNN by Alex Krizhevsky et al. won the ImageNet Large Scale Visual Recognition Challenge 2012. It was an early catalytic event for the AI boom. Compared to the training of CNNs using GPUs, not much attention was given to CPU. (Viebke et al 2019) parallelizes CNN by thread- and SIMD-level parallelism that is available on the Intel Xeon Phi. == Distinguishing features == In the past, traditional multilayer perceptron (MLP) models were used for image recognition. However, the full connectivity between nodes caused the curse of dimensionality, and was computationally intractable with higher-resolution images. A 1000×1000-pixel image with RGB color channels has 3 million weights per fully-connected neuron, which is too high to feasibly process efficiently at scale. For example, in CIFAR-10, images are only of size 32×32×3 (32 wide, 32 high, 3 color channels), so a single fully connected neuron in the first hidden layer of a regular neural network would have 32*32*3 = 3,072 weights. A 200×200 image, however, would lead to neurons that have 200*200*3 = 120,000 weights. Also, such network architecture does not take into account the spatial structure of data, treating input pixels which are far apart in the same way as pixels that are close together. This ignores locality of reference in data with a grid-topology (such as images), both computationally and semantically. Thus, full connectivity of neurons is wasteful for purposes such as image recognition that are dominated by spatially local input patterns. Convolutional neural networks are variants of multilayer perceptrons, designed to emulate the behavior of a visual cortex. These models mitigate the challenges posed by the MLP architecture by exploiting the strong spatially local correlation

present in natural images. As opposed to MLPs, CNNs have the following distinguishing features: 3D volumes of neurons. The layers of a CNN have neurons arranged in 3 dimensions: width, height and depth. Each neuron inside a convolutional layer is connected to only a small region of the layer before it, called a receptive field. Distinct types of layers, both locally and completely connected, are stacked to form a CNN architecture. Local connectivity: following the concept of receptive fields, CNNs exploit spatial locality by enforcing a local connectivity pattern between neurons of adjacent layers. The architecture thus ensures that the learned "filters" produce the strongest response to a spatially local input pattern. Stacking many such layers leads to nonlinear filters that become increasingly global (i.e. responsive to a larger region of pixel space) so that the network first creates representations of small parts of the input, then from them assembles representations of larger areas. Shared weights: In CNNs, each filter is replicated across the entire visual field. These replicated units share the same parameterization (weight vector and bias) and form a feature map. This means that all the neurons in a given convolutional layer respond to the same feature within their specific response field. Replicating units in this way allows for the resulting activation map to be equivariant under shifts of the locations of input features in the visual field, i.e. they grant translational equivariance—given that the layer has a stride of one. Pooling: In a CNN's pooling layers, feature maps are divided into rectangular sub-regions, and the features in each rectangle are independently down-sampled to a single value, commonly by taking their average or maximum value. In addition to reducing the sizes of feature maps, the pooling operation grants a degree of local translational invariance to the features contained therein, allowing the CNN to be more robust to variations in their positions. Together, these properties allow CNNs to achieve better generalization on vision problems. Weight sharing dramatically reduces the number of free parameters learned, thus lowering the memory requirements for running the network and allowing the training of larger, more powerful networks. == Building blocks == A CNN architecture is formed by a stack of distinct layers that transform the input volume into an output volume (e.g. holding the class scores) through a differentiable function. A few distinct types of layers are commonly used. These are further discussed below. === Convolutional layer === The convolutional layer is the core building block of a CNN. The layer's parameters consist of a set of learnable filters (or kernels), which have a small receptive field, but extend through the full depth of the input volume. During the forward pass, each filter is convolved across the width and height of the input volume, computing the dot product between the filter entries and the input, producing a 2-dimensional activation map of that filter. As a result, the network learns filters that activate when it detects some specific type of feature at some spatial position in the input. Stacking the activation maps for all filters along the depth dimension forms the full output volume of the convolution layer. Every entry in the output volume can thus also be interpreted as an output of a neuron that looks at a small region in the input. Each entry in an activation map use the same set of parameters that define the filter. Self-supervised learning has been adapted for use in convolutional layers by using sparse patches with a high-mask ratio and a global response normalization layer. ==== Local connectivity ==== When dealing with high-dimensional inputs such as images, it is impractical to connect neurons to all neurons in the previous volume because such a network architecture does not take the spatial structure of the data into account. Convolutional networks exploit spatially local correlation by enforcing a sparse local connectivity pattern between neurons of adjacent layers: each neuron is connected to only a small region of the input volume. The extent of this connectivity is a hyperparameter called the receptive field of the neuron. The connections are local in space (along width and height), but always extend along the entire depth of the input volume. Such an

architecture ensures that the learned filters produce the strongest response to a spatially local input pattern. ==== Spatial arrangement ==== Three hyperparameters control the size of the output volume of the convolutional layer: the depth, stride, and padding size: The depth of the output volume controls the number of neurons in a layer that connect to the same region of the input volume. These neurons learn to activate for different features in the input. For example, if the first convolutional layer takes the raw image as input, then different neurons along the depth dimension may activate in the presence of various oriented edges, or blobs of color. Stride controls how depth columns around the width and height are allocated. If the stride is 1, then we move the filters one pixel at a time. This leads to heavily overlapping receptive fields between the columns, and to large output volumes. For any integer S > 0 , {\textstyle S>0,} a stride S means that the filter is translated S units at a time per output. In practice, S ≥ 3 {\textstyle S\geq 3} is rare. A greater stride means smaller overlap of receptive fields and smaller spatial dimensions of the output volume. Sometimes, it is convenient to pad the input with zeros (or other values, such as the average of the region) on the border of the input volume. The size of this padding is a third hyperparameter. Padding provides control of the output volume's spatial size. In particular, sometimes it is desirable to exactly preserve the spatial size of the input volume, this is commonly referred to as "same" padding. The spatial size of the output volume is a function of the input volume size W {\displaystyle W} , the kernel field size K {\displaystyle K} of the convolutional layer neurons, the stride S {\displaystyle S} , and the amount of zero padding P {\displaystyle P} on the border. The number of neurons that "fit" in a given volume is then: W − K + 2 P S + 1. {\displaystyle {\frac {W-K+2P}{S}}+1.} If this number is not an integer, then the strides are incorrect and the neurons cannot be tiled to fit across the input volume in a symmetric way. In general, setting zero padding to be P = (K − 1) / 2 {\textstyle P=(K-1)/2} when the stride is S = 1 {\displaystyle S=1} ensures that the input volume and output volume will have the same size spatially. However, it is not always completely necessary to use all of the neurons of the previous layer. For example, a neural network designer may decide to use just a portion of padding. ==== Parameter sharing ==== A parameter sharing scheme is used in convolutional layers to control the number of free parameters. It relies on the assumption that if a patch feature is useful to compute at some spatial position, then it should also be useful to compute at other positions. Denoting a single 2-dimensional slice of depth as a depth slice, the neurons in each depth slice are constrained to use the same weights and bias. Since all neurons in a single depth slice share the same parameters, the forward pass in each depth slice of the convolutional layer can be computed as a convolution of the neuron's weights with the input volume. Therefore, it is common to refer to the sets of weights as a filter (or a kernel), which is convolved with the input. The result of this convolution is an activation map, and the set of activation maps for each different filter are stacked together along the depth dimension to produce the output volume. Parameter sharing contributes to the translation invariance of the CNN architecture. Sometimes, the parameter sharing assumption may not make sense. This is especially the case when the input images to a CNN have some specific centered structure; for which we expect completely different features to be learned on different spatial locations. One practical example is when the inputs are faces that have been centered in the image: we might expect different eye-specific or hair-specific features to be learned in different parts of the image. In that case it is common to relax the parameter sharing scheme, and instead simply call the layer a "locally connected layer". In this layer, the convolutional kernels' parameters are not shared. Instead, the network learns independent weights and biases for each spatial location. This allows each location to have its own feature-learning ability, making it better suited to handle images with distinct central structures or irregular features. === Pooling

layer === Another important concept of CNNs is pooling, which is used as a form of non-linear down-sampling. Pooling provides downsampling because it reduces the spatial dimensions (height and width) of the input feature maps while retaining the most important information. There are several non-linear functions to implement pooling, where max pooling and average pooling are the most common. Pooling aggregates information from small regions of the input creating partitions of the input feature map, typically using a fixed-size window (like 2x2) and applying a stride (often 2) to move the window across the input. Note that without using a stride greater than 1, pooling would not perform downsampling, as it would simply move the pooling window across the input one step at a time, without reducing the size of the feature map. In other words, the stride is what actually causes the downsampling by determining how much the pooling window moves over the input. Intuitively, the exact location of a feature is less important than its rough location relative to other features. This is the idea behind the use of pooling in convolutional neural networks. The pooling layer serves to progressively reduce the spatial size of the representation, to reduce the number of parameters, memory footprint and amount of computation in the network, and hence to also control overfitting. This is known as down-sampling. It is common to periodically insert a pooling layer between successive convolutional layers (each one typically followed by an activation function, such as a ReLU layer) in a CNN architecture. While pooling layers contribute to local translation invariance, they do not provide global translation invariance in a CNN, unless a form of global pooling is used. The pooling layer commonly operates independently on every depth, or slice, of the input and resizes it spatially. A very common form of max pooling is a layer with filters of size 2×2, applied with a stride of 2, which subsamples every depth slice in the input by 2 along both width and height, discarding 75% of the activations: f X , Y (S) = max a , b = 0 1 S 2 X + a , 2 Y + b . {\displaystyle f_{X,Y}(S)=\max _{a,b=0}^{1}S_{2X+a,2Y+b}.} In this case, every max operation is over 4 numbers. The depth dimension remains unchanged (this is true for other forms of pooling as well). In addition to max pooling, pooling units can use other functions, such as average pooling or ■2-norm pooling. Average pooling was often used historically but has recently fallen out of favor compared to max pooling, which generally performs better in practice. Due to the effects of fast spatial reduction of the size of the representation, there is a recent trend towards using smaller filters or discarding pooling layers altogether. ==== Channel max pooling ==== A channel max pooling (CMP) operation layer conducts the MP operation along the channel side among the corresponding positions of the consecutive feature maps for the purpose of redundant information elimination. The CMP makes the significant features gather together within fewer channels, which is important for fine-grained image classification that needs more discriminating features. Meanwhile, another advantage of the CMP operation is to make the channel number of feature maps smaller before it connects to the first fully connected (FC) layer. Similar to the MP operation, we denote the input feature maps and output feature maps of a CMP layer as $F \in R(C\times M\times N)$ and $C \in R(c\times M\times N)$, respectively, where C and c are the channel numbers of the input and output feature maps, M and N are the widths and the height of the feature maps, respectively. Note that the CMP operation only changes the channel number of the feature maps. The width and the height of the feature maps are not changed, which is different from the MP operation. See for reviews for pooling methods. === ReLU layer === ReLU is the abbreviation of rectified linear unit. It was proposed by Alston Householder in 1941, and used in CNN by Kunihiko Fukushima in 1969. ReLU applies the non-saturating activation function f (x) = max (0 , x) {\textstyle f(x)=\max(0,x)} . It effectively removes negative values from an activation map by setting them to zero. It introduces nonlinearity to the decision function and in the overall network without affecting the receptive fields of the

convolution layers. In 2011, Xavier Glorot, Antoine Bordes and Yoshua Bengio found that ReLU enables better training of deeper networks, compared to widely used activation functions prior to 2011. Other functions can also be used to increase nonlinearity, for example the saturating hyperbolic tangent f (x) = tanh ■ (x) {\displaystyle f(x)=\tanh(x)} , f (x) = | tanh ■ (x) | {\displaystyle f(x)=|\tanh(x)|} , and the sigmoid function σ (x) = (1 + e − x) − 1 {\textstyle \sigma (x)=(1+e^{-x})^{-1}} . ReLU is often preferred to other functions because it trains the neural network several times faster without a significant penalty to generalization accuracy. === Fully connected layer === After several convolutional and max pooling layers, the final classification is done via fully connected layers. Neurons in a fully connected layer have connections to all activations in the previous layer, as seen in regular (non-convolutional) artificial neural networks. Their activations can thus be computed as an affine transformation, with matrix multiplication followed by a bias offset (vector addition of a learned or fixed bias term). === Loss layer === The "loss layer", or "loss function", exemplifies how training penalizes the deviation between the predicted output of the network, and the true data labels (during supervised learning). Various loss functions can be used, depending on the specific task. The Softmax loss function is used for predicting a single class of K mutually exclusive classes. Sigmoid cross-entropy loss is used for predicting K independent probability values in [0 , 1] {\displaystyle [0,1]} . Euclidean loss is used for regressing to real-valued labels (− ∞ , ∞) {\displaystyle (-\infty ,\infty)} . == Hyperparameters == Hyperparameters are various settings that are used to control the learning process. CNNs use more hyperparameters than a standard multilayer perceptron (MLP). === Padding === Padding is the addition of (typically) 0-valued pixels on the borders of an image. This is done so that the border pixels are not undervalued (lost) from the output because they would ordinarily participate in only a single receptive field instance. The padding applied is typically one less than the corresponding kernel dimension. For example, a convolutional layer using 3x3 kernels would receive a 2-pixel pad, that is 1 pixel on each side of the image. === Stride === The stride is the number of pixels that the analysis window moves on each iteration. A stride of 2 means that each kernel is offset by 2 pixels from its predecessor. === Number of filters === Since feature map size decreases with depth, layers near the input layer tend to have fewer filters while higher layers can have more. To equalize computation at each layer, the product of feature values va with pixel position is kept roughly constant across layers. Preserving more information about the input would require keeping the total number of activations (number of feature maps times number of pixel positions) non-decreasing from one layer to the next. The number of feature maps directly controls the capacity and depends on the number of available examples and task complexity. === Filter (or kernel) size === Common filter sizes found in the literature vary greatly, and are usually chosen based on the data set. Typical filter sizes range from 1x1 to 7x7. As two famous examples, AlexNet used 3x3, 5x5, and 11x11. Inceptionv3 used 1x1, 3x3, and 5x5. The challenge is to find the right level of granularity so as to create abstractions at the proper scale, given a particular data set, and without overfitting. === Pooling type and size === Max pooling is typically used, often with a 2x2 dimension. This implies that the input is drastically downsampled, reducing processing cost. Greater pooling reduces the dimension of the signal, and may result in unacceptable information loss. Often, non-overlapping pooling windows perform best. === Dilation === Dilation involves ignoring pixels within a kernel. This reduces processing memory potentially without significant signal loss. A dilation of 2 on a 3x3 kernel expands the kernel to 5x5, while still processing 9 (evenly spaced) pixels. Specifically, the processed pixels after the dilation are the cells (1,1), (1,3), (1,5), (3,1), (3,3), (3,5), (5,1), (5,3), (5,5), where (i,j) denotes the cell of the i-th row and j-th column in the expanded 5x5 kernel. Accordingly,

dilation of 4 expands the kernel to 7x7. == Translation equivariance and aliasing == It is commonly assumed that CNNs are invariant to shifts of the input. Convolution or pooling layers within a CNN that do not have a stride greater than one are indeed equivariant to translations of the input. However, layers with a stride greater than one ignore the Nyquist–Shannon sampling theorem and might lead to aliasing of the input signal While, in principle, CNNs are capable of implementing anti-aliasing filters, it has been observed that this does not happen in practice, and therefore yield models that are not equivariant to translations. Furthermore, if a CNN makes use of fully connected layers, translation equivariance does not imply translation invariance, as the fully connected layers are not invariant to shifts of the input. One solution for complete translation invariance is avoiding any down-sampling throughout the network and applying global average pooling at the last layer. Additionally, several other partial solutions have been proposed, such as anti-aliasing before downsampling operations, spatial transformer networks, data augmentation, subsampling combined with pooling, and capsule neural networks. == Evaluation == The accuracy of the final model is typically estimated on a sub-part of the dataset set apart at the start, often called a test set. Alternatively, methods such as k-fold cross-validation are applied. Other strategies include using conformal prediction. == Regularization methods == Regularization is a process of introducing additional information to solve an ill-posed problem or to prevent overfitting. CNNs use various types of regularization. === Empirical === ==== Dropout ==== Because networks have so many parameters, they are prone to overfitting. One method to reduce overfitting is dropout, introduced in 2014. At each training stage, individual nodes are either "dropped out" of the net (ignored) with probability 1 − p {\displaystyle 1-p} or kept with probability p {\displaystyle p} , so that a reduced network is left; incoming and outgoing edges to a dropped-out node are also removed. Only the reduced network is trained on the data in that stage. The removed nodes are then reinserted into the network with their original weights. In the training stages, p {\displaystyle p} is usually 0.5; for input nodes, it is typically much higher because information is directly lost when input nodes are ignored. At testing time after training has finished, we would ideally like to find a sample average of all possible 2 n {\displaystyle 2^{n}} dropped-out networks; unfortunately this is unfeasible for large values of n {\displaystyle n} . However, we can find an approximation by using the full network with each node's output weighted by a factor of p {\displaystyle p} , so the expected value of the output of any node is the same as in the training stages. This is the biggest contribution of the dropout method: although it effectively generates 2 n {\displaystyle 2^{n}} neural nets, and as such allows for model combination, at test time only a single network needs to be tested. By avoiding training all nodes on all training data, dropout decreases overfitting. The method also significantly improves training speed. This makes the model combination practical, even for deep neural networks. The technique seems to reduce node interactions, leading them to learn more robust features that better generalize to new data. ==== DropConnect ==== DropConnect is the generalization of dropout in which each connection, rather than each output unit, can be dropped with probability 1 − p {\displaystyle 1-p} . Each unit thus receives input from a random subset of units in the previous layer. DropConnect is similar to dropout as it introduces dynamic sparsity within the model, but differs in that the sparsity is on the weights, rather than the output vectors of a layer. In other words, the fully connected layer with DropConnect becomes a sparsely connected layer in which the connections are chosen at random during the training stage. ==== Stochastic pooling ==== A major drawback to dropout is that it does not have the same benefits for convolutional layers, where the neurons are not fully connected. Even before dropout, in 2013 a technique called stochastic pooling, the conventional deterministic pooling operations were replaced with a stochastic

procedure, where the activation within each pooling region is picked randomly according to a multinomial distribution, given by the activities within the pooling region. This approach is free of hyperparameters and can be combined with other regularization approaches, such as dropout and data augmentation. An alternate view of stochastic pooling is that it is equivalent to standard max pooling but with many copies of an input image, each having small local deformations. This is similar to explicit elastic deformations of the input images, which delivers excellent performance on the MNIST data set. Using stochastic pooling in a multilayer model gives an exponential number of deformations since the selections in higher layers are independent of those below. ==== Artificial data ==== Because the degree of model overfitting is determined by both its power and the amount of training it receives, providing a convolutional network with more training examples can reduce overfitting. Because there is often not enough available data to train, especially considering that some part should be spared for later testing, two approaches are to either generate new data from scratch (if possible) or perturb existing data to create new ones. The latter one is used since mid-1990s. For example, input images can be cropped, rotated, or rescaled to create new examples with the same labels as the original training set. === Explicit === ==== Early stopping ==== One of the simplest methods to prevent overfitting of a network is to simply stop the training before overfitting has had a chance to occur. It comes with the disadvantage that the learning process is halted. ==== Number of parameters ==== Another simple way to prevent overfitting is to limit the number of parameters, typically by limiting the number of hidden units in each layer or limiting network depth. For convolutional networks, the filter size also affects the number of parameters. Limiting the number of parameters restricts the predictive power of the network directly, reducing the complexity of the function that it can perform on the data, and thus limits the amount of overfitting. This is equivalent to a "zero norm". ==== Weight decay ==== A simple form of added regularizer is weight decay, which simply adds an additional error, proportional to the sum of weights (L1 norm) or squared magnitude (L2 norm) of the weight vector, to the error at each node. The level of acceptable model complexity can be reduced by increasing the proportionality constant('alpha' hyperparameter), thus increasing the penalty for large weight vectors. L2 regularization is the most common form of regularization. It can be implemented by penalizing the squared magnitude of all parameters directly in the objective. The L2 regularization has the intuitive interpretation of heavily penalizing peaky weight vectors and preferring diffuse weight vectors. Due to multiplicative interactions between weights and inputs this has the useful property of encouraging the network to use all of its inputs a little rather than some of its inputs a lot. L1 regularization is also common. It makes the weight vectors sparse during optimization. In other words, neurons with L1 regularization end up using only a sparse subset of their most important inputs and become nearly invariant to the noisy inputs. L1 with L2 regularization can be combined; this is called elastic net regularization. ==== Max norm constraints ==== Another form of regularization is to enforce an absolute upper bound on the magnitude of the weight vector for every neuron and use projected gradient descent to enforce the constraint. In practice, this corresponds to performing the parameter update as normal, and then enforcing the constraint by clamping the weight vector w → {\displaystyle {\vec {w}}} of every neuron to satisfy ■ w → ■ 2 < c {\displaystyle \|{\vec {w}}\|_{2}<c} . Typical values of c {\displaystyle c} are order of 3–4. Some papers report improvements when using this form of regularization. == Hierarchical coordinate frames == Pooling loses the precise spatial relationships between high-level parts (such as nose and mouth in a face image). These relationships are needed for identity recognition. Overlapping the pools so that each feature occurs in multiple pools, helps retain the information. Translation alone cannot extrapolate the

understanding of geometric relationships to a radically new viewpoint, such as a different orientation or scale. On the other hand, people are very good at extrapolating; after seeing a new shape once they can recognize it from a different viewpoint. An earlier common way to deal with this problem is to train the network on transformed data in different orientations, scales, lighting, etc. so that the network can cope with these variations. This is computationally intensive for large data-sets. The alternative is to use a hierarchy of coordinate frames and use a group of neurons to represent a conjunction of the shape of the feature and its pose relative to the retina. The pose relative to the retina is the relationship between the coordinate frame of the retina and the intrinsic features' coordinate frame. Thus, one way to represent something is to embed the coordinate frame within it. This allows large features to be recognized by using the consistency of the poses of their parts (e.g. nose and mouth poses make a consistent prediction of the pose of the whole face). This approach ensures that the higher-level entity (e.g. face) is present when the lower-level (e.g. nose and mouth) agree on its prediction of the pose. The vectors of neuronal activity that represent pose ("pose vectors") allow spatial transformations modeled as linear operations that make it easier for the network to learn the hierarchy of visual entities and generalize across viewpoints. This is similar to the way the human visual system imposes coordinate frames in order to represent shapes. == Applications == === Image recognition === CNNs are often used in image recognition systems. In 2012, an error rate of 0.23% on the MNIST database was reported. Another paper on using CNN for image classification reported that the learning process was "surprisingly fast"; in the same paper, the best published results as of 2011 were achieved in the MNIST database and the NORB database. Subsequently, a similar CNN called AlexNet won the ImageNet Large Scale Visual Recognition Challenge 2012. When applied to facial recognition, CNNs achieved a large decrease in error rate. Another paper reported a 97.6% recognition rate on "5,600 still images of more than 10 subjects". CNNs were used to assess video quality in an objective way after manual training; the resulting system had a very low root mean square error. The ImageNet Large Scale Visual Recognition Challenge is a benchmark in object classification and detection, with millions of images and hundreds of object classes. In the ILSVRC 2014, a large-scale visual recognition challenge, almost every highly ranked team used CNN as their basic framework. The winner GoogLeNet (the foundation of DeepDream) increased the mean average precision of object detection to 0.439329, and reduced classification error to 0.06656, the best result to date. Its network applied more than 30 layers. That performance of convolutional neural networks on the ImageNet tests was close to that of humans. The best algorithms still struggle with objects that are small or thin, such as a small ant on a stem of a flower or a person holding a quill in their hand. They also have trouble with images that have been distorted with filters, an increasingly common phenomenon with modern digital cameras. By contrast, those kinds of images rarely trouble humans. Humans, however, tend to have trouble with other issues. For example, they are not good at classifying objects into fine-grained categories such as the particular breed of dog or species of bird, whereas convolutional neural networks handle this. In 2015, a many-layered CNN demonstrated the ability to spot faces from a wide range of angles, including upside down, even when partially occluded, with competitive performance. The network was trained on a database of 200,000 images that included faces at various angles and orientations and a further 20 million images without faces. They used batches of 128 images over 50,000 iterations. === Video analysis === Compared to image data domains, there is relatively little work on applying CNNs to video classification. Video is more complex than images since it has another (temporal) dimension. However, some extensions of CNNs into the video domain have been explored. One approach is to treat space and time as

equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two convolutional neural networks, one for the spatial and one for the temporal stream. Long short-term memory (LSTM) recurrent units are typically incorporated after the CNN to account for inter-frame or inter-clip dependencies. Unsupervised learning schemes for training spatio-temporal features have been introduced, based on Convolutional Gated Restricted Boltzmann Machines and Independent Subspace Analysis. Its application can be seen in text-to-video model. === Natural language processing === CNNs have also been explored for natural language processing. CNN models are effective for various NLP problems and achieved excellent results in semantic parsing, search query retrieval, sentence modeling, classification, prediction and other traditional NLP tasks. Compared to traditional language processing methods such as recurrent neural networks, CNNs can represent different contextual realities of language that do not rely on a series-sequence assumption, while RNNs are better suitable when classical time series modeling is required. === Animal behavior detection === CNNs have been applied in ecological and behavioral research to automatically detect and quantify animal behavior from visual data, enabling identification of animals, tracking of individuals, estimation of pose, and classification of specific actions such as feeding, and social interactions. Combined with multi-object tracking and temporal modeling, these systems can extract behavioral sequences over extended recordings, reducing reliance on manual annotation and increasing throughput for studies of individual variation, social networks, and collective dynamics. === Anomaly detection === A CNN with 1-D convolutions was used on time series in the frequency domain (spectral residual) by an unsupervised model to detect anomalies in the time domain. === Drug discovery === CNNs have been used in drug discovery. Predicting the interaction between molecules and biological proteins can identify potential treatments. In 2015, Atomwise introduced AtomNet, the first deep learning neural network for structure-based drug design. The system trains directly on 3-dimensional representations of chemical interactions. Similar to how image recognition networks learn to compose smaller, spatially proximate features into larger, complex structures, AtomNet discovers chemical features, such as aromaticity, sp3 carbons, and hydrogen bonding. Subsequently, AtomNet was used to predict novel candidate biomolecules for multiple disease targets, most notably treatments for the Ebola virus and multiple sclerosis. === Checkers game === CNNs have been used in the game of checkers. From 1999 to 2001, Fogel and Chellapilla published papers showing how a convolutional neural network could learn to play checkers using co-evolution. The learning process did not use prior human professional games, but rather focused on a minimal set of information contained in the checkerboard: the location and type of pieces, and the difference in number of pieces between the two sides. Ultimately, the program (Blondie24) was tested on 165 games against players and ranked in the highest 0.4%. It also earned a win against the program Chinook at its "expert" level of play. === Go === CNNs have been used in computer Go. In December 2014, Clark and Storkey published a paper showing that a CNN trained by supervised learning from a database of human professional games could outperform GNU Go and win some games against Monte Carlo tree search Fuego 1.1 in a fraction of the time it took Fuego to play. Later it was announced that a large 12-layer convolutional neural network had correctly predicted the professional move in 55% of positions, equalling the accuracy of a 6 dan human player. When the trained convolutional network was used directly to play games of Go, without any search, it beat the traditional search program GNU Go in 97% of games, and matched the performance of the Monte Carlo tree search program Fuego simulating ten thousand playouts (about a million positions) per move. A couple of CNNs for choosing moves to try ("policy network") and evaluating

positions ("value network") driving MCTS were used by AlphaGo, the first to beat the best human player at the time. === Time series forecasting === Recurrent neural networks are generally considered the best neural network architectures for time series forecasting (and sequence modeling in general), but recent studies show that convolutional networks can perform comparably or even better. Dilated convolutions might enable one-dimensional convolutional neural networks to effectively learn time series dependences. Convolutions can be implemented more efficiently than RNN-based solutions, and they do not suffer from vanishing (or exploding) gradients. Convolutional networks can provide an improved forecasting performance when there are multiple similar time series to learn from. CNNs can also be applied to further tasks in time series analysis (e.g., time series classification or quantile forecasting). === Cultural heritage and 3D-datasets === As archaeological findings such as clay tablets with cuneiform writing are increasingly acquired using 3D scanners, benchmark datasets are becoming available, including HeiCuBeDa providing almost 2000 normalized 2-D and 3-D datasets prepared with the GigaMesh Software Framework. So curvature-based measures are used in conjunction with geometric neural networks (GNNs), e.g. for period classification of those clay tablets being among the oldest documents of human history. == Fine-tuning == For many applications, training data is not very available. Convolutional neural networks usually require a large amount of training data in order to avoid overfitting. A common technique is to train the network on a larger data set from a related domain. Once the network parameters have converged an additional training step is performed using the in-domain data to fine-tune the network weights, this is known as transfer learning. Furthermore, this technique allows convolutional network architectures to successfully be applied to problems with tiny training sets. == Human interpretable explanations == End-to-end training and prediction are common practice in computer vision. However, human interpretable explanations are required for critical systems such as self-driving cars. With recent advances in visual salience, spatial attention, and temporal attention, the most critical spatial regions/temporal instants could be visualized to justify the CNN predictions. == Related architectures == === Deep Q-networks === A deep Q-network (DQN) is a type of deep learning model that combines a deep neural network with Q-learning, a form of reinforcement learning. Unlike earlier reinforcement learning agents, DQNs that utilize CNNs can learn directly from high-dimensional sensory inputs via reinforcement learning. Preliminary results were presented in 2014, with an accompanying paper in February 2015. The research described an application to Atari 2600 gaming. Other deep reinforcement learning models preceded it. === Deep belief networks === Convolutional deep belief networks (CDBN) have structure very similar to convolutional neural networks and are trained similarly to deep belief networks. Therefore, they exploit the 2D structure of images, like CNNs do, and make use of pre-training like deep belief networks. They provide a generic structure that can be used in many image and signal processing tasks. Benchmark results on standard image datasets like CIFAR have been obtained using CDBNs. === Neural abstraction pyramid === The feed-forward architecture of convolutional neural networks was extended in the neural abstraction pyramid by lateral and feedback connections. The resulting recurrent convolutional network allows for the flexible incorporation of contextual information to iteratively resolve local ambiguities. In contrast to previous models, image-like outputs at the highest resolution were generated, e.g., for semantic segmentation, image reconstruction, and object localization tasks. == Notable libraries == Caffe: A library for convolutional neural networks. Created by the Berkeley Vision and Learning Center (BVLC). It supports both CPU and GPU. Developed in C++, and has Python and MATLAB wrappers. Deeplearning4j: Deep learning in Java and Scala on multi-GPU-enabled Spark. A general-purpose

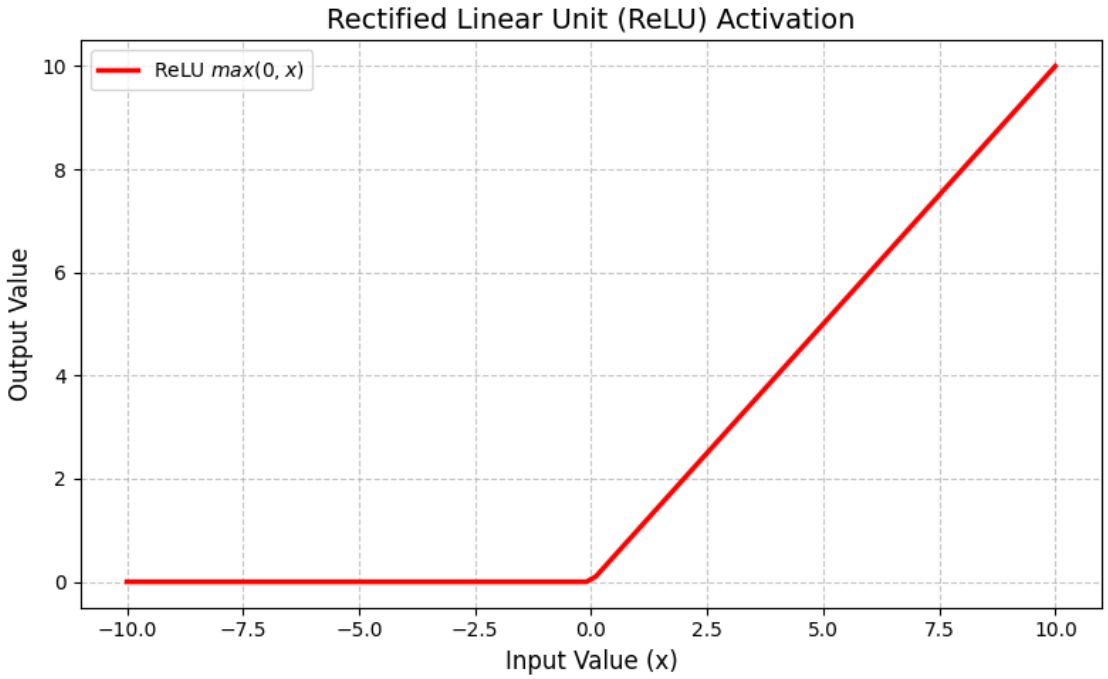
deep learning library for the JVM production stack running on a C++ scientific computing engine. Allows the creation of custom layers. Integrates with Hadoop and Kafka. Dlib: A toolkit for making real world machine learning and data analysis applications in C++. Microsoft Cognitive Toolkit: A deep learning toolkit written by Microsoft with several unique features enhancing scalability over multiple nodes. It supports full-fledged interfaces for training in C++ and Python and with additional support for model inference in C# and Java. TensorFlow: Apache 2.0-licensed Theano-like library with support for CPU, GPU, Google's proprietary tensor processing unit (TPU), and mobile devices. Theano: The reference deep-learning library for Python with an API largely compatible with the popular NumPy library. Allows user to write symbolic mathematical expressions, then automatically generates their derivatives, saving the user from having to code gradients or backpropagation. These symbolic expressions are automatically compiled to CUDA code for a fast, on-the-GPU implementation. Torch: A scientific computing framework with wide support for machine learning algorithms, written in C and Lua. == See also == Attention (machine learning) Convolution Deep learning Natural-language processing Neocognitron Scale-invariant feature transform Time delay neural network Vision processing unit == Notes == == References == == External links == CS231n: Convolutional Neural Networks for Visual Recognition — Andrej Karpathy's Stanford computer science course on CNNs in computer vision vdumoulin/conv_arithmetic: A technical report on convolution arithmetic in the context of deep learning. Animations of convolutions.

The Mathematical Operations of a CNN

A convolution is a specialized linear operation. For a 2D image I and a 2D kernel K, the convolution operation is defined as:

$$S(i,j) = (I * K)(i,j) = \Sigma_m \Sigma_n I(m,n) K(i-m, j-n)$$

CNNs use activation functions to introduce non-linearity. The most common is the Rectified Linear Unit (ReLU).


Rectified Linear Unit (ReLU) Activation
ReLU $max(0, x)$
Output Value
Input Value (x)
10
8
6
4
2
0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

1.3 Computer Vision & Object Detection

Object detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Well-researched domains of object detection include face detection and pedestrian detection. Object detection has applications in many areas of computer vision, including image retrieval and video surveillance. == Uses == It is widely used in computer vision tasks such as image annotation, vehicle counting, activity recognition, face detection, face recognition, video object co-segmentation. It is also used in tracking objects, for example tracking a ball during a football match, tracking movement of a cricket bat, or tracking a person in a video. Often, the test images are sampled from a different data distribution, making the object detection task significantly more difficult. To address the challenges caused by the domain gap between training and test data, many unsupervised domain adaptation approaches have been proposed. A simple and straightforward solution for reducing the domain gap is to apply an image-to-image translation approach, such as cycle-GAN. Among other uses, cross-domain object detection is applied in autonomous driving, where models can be trained on a vast amount of video game scenes, since the labels can be generated without manual labor. == Concept == Every object class has its own special features that help in classifying the class – for example all circles are round. Object class detection uses these special features. For example, when looking for circles, objects that are at a particular distance from a point (i.e. the center) are sought. Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for face identification where eyes, nose, and lips can be found and features like skin color and distance between eyes can be found. == Benchmarks == For object localization, true positive is often measured by the thresholded intersection over union. For example, if there is a traffic sign in the image, with a bounding box drawn by a human ("ground truth label"), then a neural network has detected the traffic sign (a true positive) at 0.5 threshold iff it has drawn a bounding box whose IoU with the ground truth is above 0.5. Otherwise, the bounding box is a false positive. If there is only a single ground truth bounding box, but multiple predictions, then the IoU of each prediction is calculated. The prediction with the highest IoU is a true positive if it is above threshold, else it is a false positive. All other predicted bounding boxes are false positives. If there is no prediction with an IoU above the threshold, then the ground truth label has a false negative. For simultaneous object localization and classification, a true positive is one where the class label is correct, and the bounding box has an IoU exceeding the threshold. Simultaneous object localization and classification is benchmarked by the mean average precision (mAP). The average precision (AP) of the network for a class of objects is the area under the precision-recall curve as the confidence threshold is varied. The mAP is the average of AP over all classes. == Methods == Methods for object detection generally fall into either neural network-based or non-neural approaches. For non-neural approaches, it becomes necessary to first define features using one of the methods below, then using a technique such as support vector machine (SVM) to do the classification. On the other hand, neural techniques are able to do end-to-end object detection without specifically defining features, and are typically based on convolutional neural networks (CNN). Non-neural approaches: Viola–Jones object detection framework based on Haar features Scale-invariant feature transform (SIFT) Histogram of oriented gradients (HOG) features Neural network approaches: OverFeat. Region Proposals (R-CNN, Fast R-CNN, Faster R-CNN, cascade R-CNN.) You Only Look Once (YOLO). Single Shot MultiBox Detector (SSD) Single-Shot

Refinement Neural Network for Object Detection (RefineDet) Retina-Net Deformable convolutional networks == See also == Feature detection (computer vision) Moving object detection Small object detection Outline of object recognition Teknomo–Fernandez algorithm == References == == Further reading == Zou, Zhengxia; Chen, Keyan; Shi, Zhenwei; Guo, Yuhong; Ye, Jieping (March 2023). "Object Detection in 20 Years: A Survey". Proceedings of the IEEE. 111 (3): 257–276. doi:10.1109/JPROC.2023.3238524. ISSN 0018-9219. == External links == Joshi, Snehal (2023-10-24). "Top Object Detection Models". hitechbpo.com. Weng, Lilian (2017-10-29). "Object Detection for Dummies Part 1: Gradient Vector, HOG, and SS". lilianweng.github.io. Retrieved 2024-09-11. Weng, Lilian (2017-12-15). "Object Detection for Dummies Part 2: CNN, DPM and Overfeat". lilianweng.github.io. Retrieved 2024-09-11. Weng, Lilian (2017-12-31). "Object Detection for Dummies Part 3: R-CNN Family". lilianweng.github.io. Retrieved 2024-09-11. Weng, Lilian (2018-12-27). "Object Detection Part 4: Fast Detection Models". lilianweng.github.io. Retrieved 2024-09-11. Multiple object class detection Spatio-temporal action localization Online Object Detection Demo Video object detection and co-segmentation

Part II: Face Detection Architecture (YuNet)

2.1 Introduction to YuNet

YuNet is a lightweight, high-performance face detector designed for edge computing devices. Unlike early models like Viola-Jones (Haar Cascades) which rely on manually engineered features, YuNet learns robust features directly from raw pixels using a CNN.

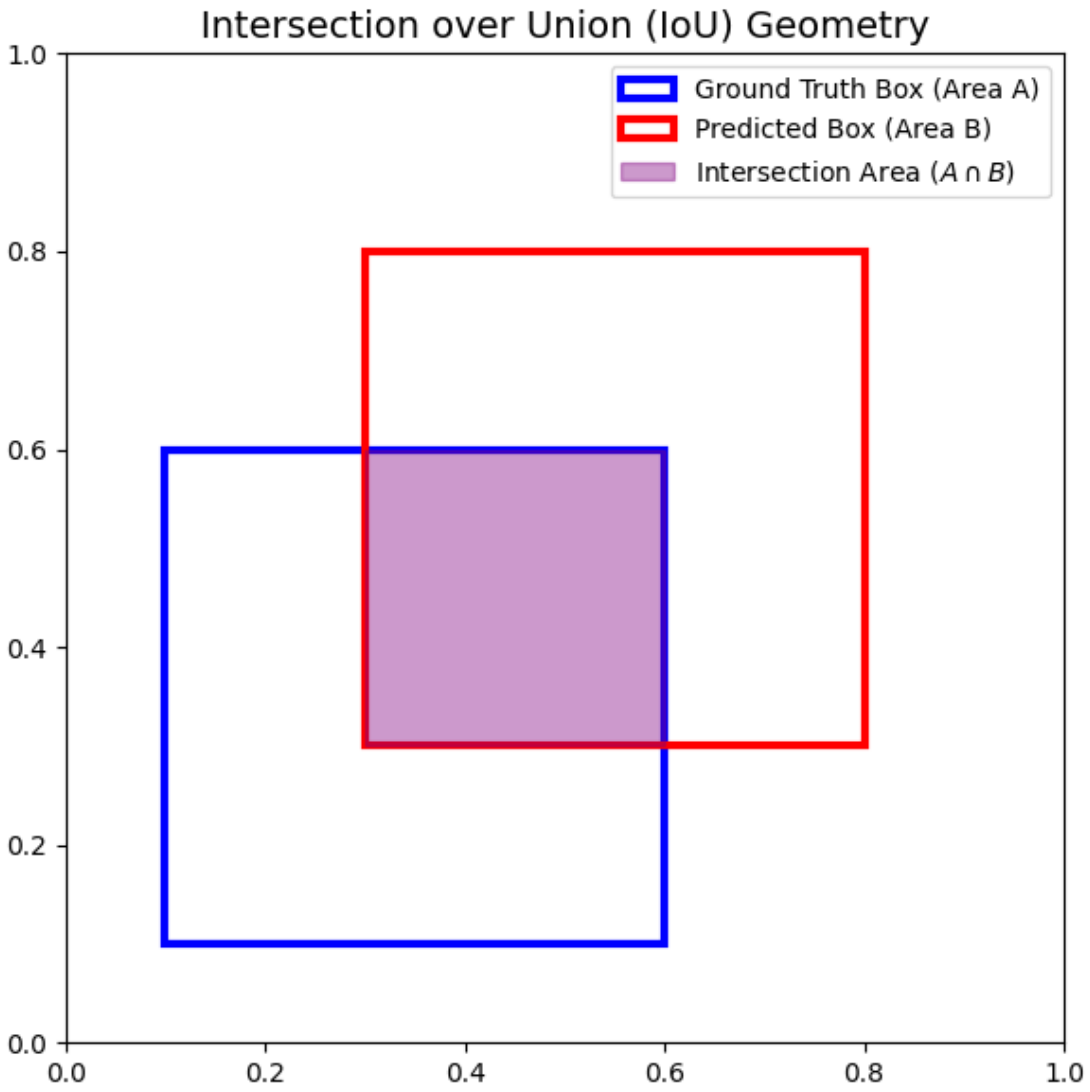
2.2 Anchor-Free Design

Traditional detectors (like SSD or YOLOv3) use 'anchors'—pre-defined bounding boxes of various scales and aspect ratios scattered across the image. The model then predicts offsets for these anchors. Anchor-free models, however, predict the center point of objects directly. This removes the need for complex anchor tuning and significantly reduces the computational overhead, making YuNet incredibly fast.

2.3 Intersection over Union (IoU) and EIoU Loss

To train the detector, the network needs to know how 'wrong' its predictions are. The standard metric is IoU.

IoU = Area of Overlap / Area of Union



However, standard IoU is zero if the boxes do not overlap, meaning no gradient flows back through the network. YuNet utilizes Extended IoU (EIoU) which calculates the distance between the center points of the boxes, the width differences, and the height differences. This ensures the network continues to learn even when predictions are completely outside the ground truth.

Part III: Face Recognition Architecture (SFace)

3.1 Evolution of Facial Recognition

A facial recognition system is a technology potentially capable of matching a human face from a digital image or a video frame against a database of faces. Such a system is typically employed to authenticate users through ID verification services, and works by pinpointing and measuring facial features from a given image. Development on similar systems began in the 1960s as a form of computer application. Since their inception, facial recognition systems have seen wider uses in recent times on smartphones and in other forms of technology, such as robotics. Because computerized facial recognition involves the measurement of a human's physiological characteristics, facial recognition systems are categorized as biometrics. Although the accuracy of facial recognition systems as a biometric technology is lower than iris recognition, fingerprint image acquisition, palm recognition or voice recognition, it is widely adopted due to its contactless process. Facial recognition systems have been deployed in advanced human–computer interaction, video surveillance, law enforcement, passenger screening, decisions on employment and housing, and automatic indexing of images. Facial recognition systems are employed throughout the world today by governments and private companies. Their effectiveness varies, and some systems have previously been scrapped because of their ineffectiveness. The use of facial recognition systems has also raised controversy, with claims that the systems violate citizens' privacy, commonly make incorrect identifications, encourage gender norms and racial profiling, and do not protect important biometric data. The appearance of synthetic media such as deepfakes has also raised concerns about its security. These claims have led to the ban of facial recognition systems in several cities in the United States. Growing societal concerns led social networking company Meta Platforms to shut down its Facebook facial recognition system in 2021, deleting the face-scan data of more than one billion users. The change represented one of the largest shifts in facial recognition usage in the technology's history. IBM also stopped offering facial recognition technology due to similar concerns. == History of facial recognition technology == Automated facial recognition was pioneered in the 1960s by Woody Bledsoe, Helen Chan Wolf, and Charles Bisson, whose work focused on teaching computers to recognize human faces. Their early facial recognition project was dubbed "man-machine" because a human first needed to establish the coordinates of facial features in a photograph before they could be used by a computer for recognition. Using a graphics tablet, a human would pinpoint facial features coordinates, such as the pupil centers, the inside and outside corners of eyes, and the widows peak in the hairline. The coordinates were used to calculate 20 individual distances, including the width of the mouth and of the eyes. A human could process about 40 pictures an hour, building a database of these computed distances. A computer would then automatically compare the distances for each photograph, calculate the difference between the distances, and return the closed records as a possible match. In 1970, Takeo Kanade publicly demonstrated a face-matching system that located anatomical features such as the chin and calculated the distance ratio between facial features without human intervention. Later tests revealed that the system could not always reliably identify facial features. Nonetheless, interest in the subject grew and in 1977 Kanade published the first detailed book on facial recognition technology. In 1993, the Defense Advanced Research Project Agency (DARPA) and the Army

Research Laboratory (ARL) established the face recognition technology program FERET to develop "automatic face recognition capabilities" that could be employed in a productive real life environment "to assist security, intelligence, and law enforcement personnel in the performance of their duties." Face recognition systems that had been trialled in research labs were evaluated. The FERET tests found that while the performance of existing automated facial recognition systems varied, a handful of existing methods could viably be used to recognize faces in still images taken in a controlled environment. The FERET tests spawned three US companies that sold automated facial recognition systems. Vision Corporation and Miros Inc were founded in 1994, by researchers who used the results of the FERET tests as a selling point. Viisage Technology was established by an identification card defense contractor in 1996 to commercially exploit the rights to the facial recognition algorithm developed by Alex Pentland at MIT. Following the 1993 FERET face-recognition vendor test, the Department of Motor Vehicles (DMV) offices in West Virginia and New Mexico became the first DMV offices to use automated facial recognition systems to prevent people from obtaining multiple driving licenses using different names. Driver's licenses in the United States were at that point a commonly accepted form of photo identification. DMV offices across the United States were undergoing a technological upgrade and were in the process of establishing databases of digital ID photographs. This enabled DMV offices to deploy the facial recognition systems on the market to search photographs for new driving licenses against the existing DMV database. DMV offices became one of the first major markets for automated facial recognition technology and introduced US citizens to facial recognition as a standard method of identification. The increase of the US prison population in the 1990s prompted U.S. states to established connected and automated identification systems that incorporated digital biometric databases, in some instances this included facial recognition. In 1999, Minnesota incorporated the facial recognition system FaceIT by Visionics into a mug shot booking system that allowed police, judges and court officers to track criminals across the state. Until the 1990s, facial recognition systems were developed primarily by using photographic portraits of human faces. Research on face recognition to reliably locate a face in an image that contains other objects gained traction in the early 1990s with the principal component analysis (PCA). The PCA method of face detection is also known as Eigenface and was developed by Matthew Turk and Alex Pentland. Turk and Pentland combined the conceptual approach of the Karhunen–Loève theorem and factor analysis, to develop a linear model. Eigenfaces are determined based on global and orthogonal features in human faces. A human face is calculated as a weighted combination of a number of Eigenfaces. Because few Eigenfaces were used to encode human faces of a given population, Turk and Pentland's PCA face detection method greatly reduced the amount of data that had to be processed to detect a face. Pentland in 1994 defined Eigenface features, including eigen eyes, eigen mouths and eigen noses, to advance the use of PCA in facial recognition. In 1997, the PCA Eigenface method of face recognition was improved upon using linear discriminant analysis (LDA) to produce Fisherfaces. LDA Fisherfaces became dominantly used in PCA feature based face recognition. While Eigenfaces were also used for face reconstruction. In these approaches no global structure of the face is calculated which links the facial features or parts. Purely feature based approaches to facial recognition were overtaken in the late 1990s by the Bochum system, which used Gabor filter to record the face features and computed a grid of the face structure to link the features. Christoph von der Malsburg and his research team at the University of Bochum developed Elastic Bunch Graph Matching in the mid-1990s to extract a face out of an image using skin segmentation. By 1997, the face detection method developed by Malsburg outperformed most other facial detection

systems on the market. The so-called "Bochum system" of face detection was sold commercially on the market as ZN-Face to operators of airports and other busy locations. The software was "robust enough to make identifications from less-than-perfect face views. It can also often see through such impediments to identification as mustaches, beards, changed hairstyles and glasses—even sunglasses". Real-time face detection in video footage became possible in 2001 with the Viola–Jones object detection framework for faces. Paul Viola and Michael Jones combined their face detection method with the Haar-like feature approach to object recognition in digital images to launch AdaBoost, the first real-time frontal-view face detector. By 2015, the Viola–Jones algorithm had been implemented using small low power detectors on handheld devices and embedded systems. Therefore, the Viola–Jones algorithm has not only broadened the practical application of face recognition systems but has also been used to support new features in user interfaces and teleconferencing. Ukraine is using the US-based Clearview AI facial recognition software to identify dead Russian soldiers. Ukraine has conducted 8,600 searches and identified the families of 582 deceased Russian soldiers. The IT volunteer section of the Ukrainian army using the software is subsequently contacting the families of the deceased soldiers to raise awareness of Russian activities in Ukraine. The main goal is to destabilise the Russian government. It can be seen as a form of psychological warfare. About 340 Ukrainian government officials in five government ministries are using the technology. It is used to catch spies that might try to enter Ukraine. Clearview AI's facial recognition database is only available to government agencies who may only use the technology to assist in the course of law enforcement investigations or in connection with national security. The software was donated to Ukraine by Clearview AI. Russia is thought to be using it to find anti-war activists. Clearview AI was originally designed for US law enforcement. Using it in war raises new ethical concerns. One London based surveillance expert, Stephen Hare, is concerned it might make the Ukrainians appear inhuman: "Is it actually working? Or is it making [Russians] say: 'Look at these lawless, cruel Ukrainians, doing this to our boys'?" == Techniques for face recognition == While humans can recognize faces without much effort, facial recognition is a challenging pattern recognition problem in computing. Facial recognition systems attempt to identify a human face, which is three-dimensional and changes in appearance with lighting and facial expression, based on its two-dimensional image. To accomplish this computational task, facial recognition systems perform four steps. First face detection is used to segment the face from the image background. In the second step the segmented face image is aligned to account for face pose, image size and photographic properties, such as illumination and grayscale. The purpose of the alignment process is to enable the accurate localization of facial features in the third step, the facial feature extraction. Features such as eyes, nose and mouth are pinpointed and measured in the image to represent the face. The so established feature vector of the face is then, in the fourth step, matched against a database of faces. === Traditional === Some face recognition algorithms identify facial features by extracting landmarks, or features, from an image of the subject's face. For example, an algorithm may analyze the relative position, size, and/or shape of the eyes, nose, cheekbones, and jaw. These features are then used to search for other images with matching features. Other algorithms normalize a gallery of face images and then compress the face data, only saving the data in the image that is useful for face recognition. A probe image is then compared with the face data. One of the earliest successful systems is based on template matching techniques applied to a set of salient facial features, providing a sort of compressed face representation. Recognition algorithms can be divided into two main approaches: geometric, which looks at distinguishing features, or photo-metric, which is a statistical approach that distills an image

into values and compares the values with templates to eliminate variances. Some classify these algorithms into two broad categories: holistic and feature-based models. The former attempts to recognize the face in its entirety while the feature-based subdivide into components such as according to features and analyze each as well as its spatial location with respect to other features. Popular recognition algorithms include principal component analysis using eigenfaces, linear discriminant analysis, elastic bunch graph matching using the Fisherface algorithm, the hidden Markov model, the multilinear subspace learning using tensor representation, and the neuronal motivated dynamic link matching. Modern facial recognition systems make increasing use of machine learning techniques such as deep learning. === Human identification at a distance (HID) === To enable human identification at a distance (HID) low-resolution images of faces are enhanced using face hallucination. In CCTV imagery faces are often very small. But because facial recognition algorithms that identify and plot facial features require high resolution images, resolution enhancement techniques have been developed to enable facial recognition systems to work with imagery that has been captured in environments with a high signal-to-noise ratio. Face hallucination algorithms that are applied to images prior to those images being submitted to the facial recognition system use example-based machine learning with pixel substitution or nearest neighbour distribution indexes that may also incorporate demographic and age related facial characteristics. Use of face hallucination techniques improves the performance of high resolution facial recognition algorithms and may be used to overcome the inherent limitations of super-resolution algorithms. Face hallucination techniques are also used to pre-treat imagery where faces are disguised. Here the disguise, such as sunglasses, is removed and the face hallucination algorithm is applied to the image. Such face hallucination algorithms need to be trained on similar face images with and without disguise. To fill in the area uncovered by removing the disguise, face hallucination algorithms need to correctly map the entire state of the face, which may be not possible due to the momentary facial expression captured in the low resolution image. === 3-dimensional recognition === Three-dimensional face recognition technique uses 3D sensors to capture information about the shape of a face. This information is then used to identify distinctive features on the surface of a face, such as the contour of the eye sockets, nose, and chin. One advantage of 3D face recognition is that it is not affected by changes in lighting like other techniques. It can also identify a face from a range of viewing angles, including a profile view. Three-dimensional data points from a face vastly improve the precision of face recognition. 3D-dimensional face recognition research is enabled by the development of sophisticated sensors that project structured light onto the face. 3D matching technique are sensitive to expressions, therefore researchers at Technion applied tools from metric geometry to treat expressions as isometries. A new method of capturing 3D images of faces uses three tracking cameras that point at different angles; one camera will be pointing at the front of the subject, second one to the side, and third one at an angle. All these cameras will work together so it can track a subject's face in real-time and be able to face detect and recognize. === Thermal cameras === A different form of taking input data for face recognition is by using thermal cameras, by this procedure the cameras will only detect the shape of the head and it will ignore the subject accessories such as glasses, hats, or makeup. Unlike conventional cameras, thermal cameras can capture facial imagery even in low-light and nighttime conditions without using a flash and exposing the position of the camera. However, the databases for face recognition are limited. Efforts to build databases of thermal face images date back to 2004. By 2016, several databases existed, including the IIITD-PSE and the Notre Dame thermal face database. Current thermal face recognition systems are not able to reliably detect a face in a thermal image that has been taken of an outdoor

environment. In 2018, researchers from the U.S. Army Research Laboratory (ARL) developed a technique that would allow them to match facial imagery obtained using a thermal camera with those in databases that were captured using a conventional camera. Known as a cross-spectrum synthesis method due to how it bridges facial recognition from two different imaging modalities, this method synthesize a single image by analyzing multiple facial regions and details. It consists of a non-linear regression model that maps a specific thermal image into a corresponding visible facial image and an optimization issue that projects the latent projection back into the image space. ARL scientists have noted that the approach works by combining global information (i.e. features across the entire face) with local information (i.e. features regarding the eyes, nose, and mouth). According to performance tests conducted at ARL, the multi-region cross-spectrum synthesis model demonstrated a performance improvement of about 30% over baseline methods and about 5% over state-of-the-art methods. == Application == === Social media === Founded in 2013, Looksery went on to raise money for its face modification app on Kickstarter. After successful crowdfunding, Looksery launched in October 2014. The application allows video chat with others through a special filter for faces that modifies the look of users. Image augmenting applications already on the market, such as Facetune and Perfect365, were limited to static images, whereas Looksery allowed augmented reality to live videos. In late 2015 SnapChat purchased Looksery, which would then become its landmark lenses function. Snapchat filter applications use face detection technology and on the basis of the facial features identified in an image a 3D mesh mask is layered over the face. A variety of technologies attempt to fool facial recognition software by the use of anti-facial recognition masks. DeepFace is a deep learning facial recognition system created by a research group at Facebook. It identifies human faces in digital images. It employs a nine-layer neural net with over 120 million connection weights, and was trained on four million images uploaded by Facebook users. The system is said to be 97% accurate, compared to 85% for the FBI's Next Generation Identification system. TikTok's algorithm has been regarded as especially effective, but many were left to wonder at the exact programming that caused the app to be so effective in guessing the user's desired content. In June 2020, TikTok released a statement regarding the "For You" page, and how they recommended videos to users, which did not include facial recognition. In February 2021, however, TikTok agreed to a $92 million settlement to a US lawsuit which alleged that the app had used facial recognition in both user videos and its algorithm to identify age, gender and ethnicity. === ID verification === The emerging use of facial recognition is in the use of ID verification services. Many companies and others are working in the market now to provide these services to banks, ICOs, and other e-businesses. Face recognition has been leveraged as a form of biometric authentication for various computing platforms and devices; Android 4.0 "Ice Cream Sandwich" added facial recognition using a smartphone's front camera as a means of unlocking devices, while Microsoft introduced face recognition login to its Xbox 360 video game console through its Kinect accessory, as well as Windows 10 via its "Windows Hello" platform (which requires an infrared-illuminated camera). In 2017, Apple's iPhone X smartphone introduced facial recognition to the product line with its "Face ID" platform, which uses an infrared illumination system. ==== Face ID ==== Apple introduced Face ID on the flagship iPhone X as a biometric authentication successor to the Touch ID, a fingerprint based system. Face ID has a facial recognition sensor that consists of two parts: a "Romeo" module that projects more than 30,000 infrared dots onto the user's face, and a "Juliet" module that reads the pattern. The pattern is sent to a local "Secure Enclave" in the device's central processing unit (CPU) to confirm a match with the phone owner's face. The facial pattern is not accessible by Apple. The system will not work with

eyes closed, in an effort to prevent unauthorized access. The technology learns from changes in a user's appearance, and therefore works with hats, scarves, glasses, and many sunglasses, beard and makeup. It also works in the dark. This is done by using a "Flood Illuminator", which is a dedicated infrared flash that throws out invisible infrared light onto the user's face to get a 2d picture in addition to the 30,000 facial points. === Healthcare === Facial recognition algorithms can help in diagnosing some diseases using specific features on the nose, cheeks and other part of the human face. Relying on developed data sets, machine learning has been used to identify genetic abnormalities just based on facial dimensions. FRT has also been used to verify patients before surgery procedures. In March, 2022 according to a publication by Forbes, FDNA, an AI development company claimed that in the space of 10 years, they have worked with geneticists to develop a database of about 5,000 diseases and 1500 of them can be detected with facial recognition algorithms. === Deployment of FRT for availing government services === ==== India ==== In an interview, the National Health Authority chief Dr. R.S. Sharma said that facial recognition technology would be used in conjunction with Aadhaar to authenticate the identity of people seeking vaccines. Ten human rights and digital rights organizations and more than 150 individuals signed a statement by the Internet Freedom Foundation that raised alarm against the deployment of facial recognition technology in the central government's vaccination drive process. Implementation of an error-prone system without adequate legislation containing mandatory safeguards, would deprive citizens of essential services and linking this untested technology to the vaccination roll-out in India will only exclude persons from the vaccine delivery system. In July, 2021, a press release by the Government of Meghalaya stated that facial recognition technology (FRT) would be used to verify the identity of pensioners to issue a Digital Life Certificate using "Pensioner's Life Certification Verification" mobile application. The notice, according to the press release, purports to offer pensioners "a secure, easy and hassle-free interface for verifying their liveness to the Pension Disbursing Authorities from the comfort of their homes using smart phones". Mr. Jade Jeremiah Lyngdoh, a law student, sent a legal notice to the relevant authorities highlighting that "The application has been rolled out without any anchoring legislation which governs the processing of personal data and thus, lacks lawfulness and the Government is not empowered to process data." === Deployment in security services === ==== Commonwealth ==== The Australian Border Force and New Zealand Customs Service have set up an automated border processing system called SmartGate that uses face recognition, which compares the face of the traveller with the data in the e-passport microchip. All Canadian international airports use facial recognition as part of the Primary Inspection Kiosk program that compares a traveler face to their photo stored on the ePassport. This program first came to Vancouver International Airport in early 2017 and was rolled up to all remaining international airports in 2018–2019. Police forces in the United Kingdom have been trialing live facial recognition technology at public events since 2015. In May 2017, a man was arrested using an automatic facial recognition (AFR) system mounted on a van operated by the South Wales Police. Ars Technica reported that "this appears to be the first time [AFR] has led to an arrest". However, a 2018 report by Big Brother Watch found that these systems were up to 98% inaccurate. The report also revealed that two UK police forces, South Wales Police and the Metropolitan Police, were using live facial recognition at public events and in public spaces. In September 2019, South Wales Police use of facial recognition was ruled lawful. Live facial recognition has been trialled since 2016 in the streets of London and will be used on a regular basis from Metropolitan Police from beginning of 2020. In August 2020 the Court of Appeal ruled that the way the facial recognition system had been used by the South Wales Police in 2017

and 2018 violated human rights. However, by 2024 the Metropolitan Police were using the technique with a database of 16,000 suspects, leading to over 360 arrests, including rapists and someone wanted for grievous bodily harm for 8 years. They claim a false positive rate of only 1 in 6,000. The photos of those not identified by the system are deleted immediately. ==== United States ==== The U.S. Department of State operates one of the largest face recognition systems in the world with a database of 117 million American adults, with photos typically drawn from driver's license photos. Although it is still far from completion, it is being put to use in certain cities to give clues as to who was in the photo. The FBI uses the photos as an investigative tool, not for positive identification. As of 2016, facial recognition was being used to identify people in photos taken by police in San Diego and Los Angeles (not on real-time video, and only against booking photos) and use was planned in West Virginia and Dallas. In recent years Maryland has used face recognition by comparing people's faces to their driver's license photos. The system drew controversy when it was used in Baltimore to arrest unruly protesters after the death of Freddie Gray in police custody. Many other states are using or developing a similar system however some states have laws prohibiting its use. The FBI has also instituted its Next Generation Identification program to include face recognition, as well as more traditional biometrics like fingerprints and iris scans, which can pull from both criminal and civil databases. The federal Government Accountability Office criticized the FBI for not addressing various concerns related to privacy and accuracy. Starting in 2018, U.S. Customs and Border Protection deployed "biometric face scanners" at U.S. airports. Passengers taking outbound international flights can complete the check-in, security and the boarding process after getting facial images captured and verified by matching their ID photos stored on CBP's database. Images captured for travelers with U.S. citizenship will be deleted within up to 12-hours. The Transportation Security Administration (TSA) had expressed its intention to adopt a similar program for domestic air travel during the security check process in the future. The American Civil Liberties Union is one of the organizations against the program, concerning that the program will be used for surveillance purposes. In 2019, researchers reported that Immigration and Customs Enforcement (ICE) uses facial recognition software against state driver's license databases, including for some states that provide licenses to undocumented immigrants. In December 2022, 16 major domestic airports in the US started testing facial-recognition tech where kiosks with cameras are checking the photos on travelers' IDs to make sure that passengers are not impostors. In 2025, it was revealed that the New Orleans Police Department had rolled out what the ACLU's Freed Wessler called "the first known widespread effort by police in a major US city to use AI to identify people in live camera feeds for the purpose of making immediate arrests." in defiance of a 2022 city ordinance limiting the use of the technology. ==== China ==== In 2006, the "Skynet" (■■■)Project was initiated by the Chinese government to implement CCTV surveillance nationwide and as of 2018, there have been 20 million cameras, many of which are capable of real-time facial recognition, deployed across the country for this project. Some official claim that the current Skynet system can scan the entire Chinese population in one second and the world population in two seconds. In 2017, the Qingdao police was able to identify twenty-five wanted suspects using facial recognition equipment at the Qingdao International Beer Festival, one of which had been on the run for 10 years. The equipment works by recording a 15-second video clip and taking multiple snapshots of the subject. That data is compared and analyzed with images from the police department's database and within 20 minutes, the subject can be identified with a 98.1% accuracy. In 2018, Chinese police in Zhengzhou and Beijing were using smart glasses to take photos which are compared against a government database using facial recognition to identify suspects, retrieve

an address, and track people moving beyond their home areas. As of late 2017, China has deployed facial recognition and artificial intelligence technology in Xinjiang. Reporters visiting the region found surveillance cameras installed every hundred meters or so in several cities, as well as facial recognition checkpoints at areas like gas stations, shopping centers, and mosque entrances. In May 2019, Human Rights Watch reported finding Face++ code in the Integrated Joint Operations Platform (IJOP), a police surveillance app used to collect data on, and track the Uighur community in Xinjiang. Human Rights Watch released a correction to its report in June 2019 stating that the Chinese company Megvii did not appear to have collaborated on IJOP, and that the Face++ code in the app was inoperable. In February 2020, following the Coronavirus outbreak, Megvii applied for a bank loan to optimize the body temperature screening system it had launched to help identify people with symptoms of a Coronavirus infection in crowds. In its loan application documents, Megvii stated that it needed to improve the accuracy of identifying masked individuals. Many public places in China are implemented with facial recognition equipment, including railway stations, airports, tourist attractions, expos, and office buildings. In October 2019, a professor at Zhejiang Sci-Tech University sued the Hangzhou Safari Park for abusing private biometric information of customers. The safari park uses facial recognition technology to verify the identities of its Year Card holders. An estimated 300 tourist sites in China have installed facial recognition systems and use them to admit visitors. This case is reported to be the first on the use of facial recognition systems in China. In August 2020, Radio Free Asia reported that in 2019 Geng Guanjun, a citizen of Taiyuan City who had used the WeChat app by Tencent to forward a video to a friend in the United States was subsequently convicted on the charge of the crime "picking quarrels and provoking troubles". The Court documents showed that the Chinese police used a facial recognition system to identify Geng Guanjun as an "overseas democracy activist" and that China's network management and propaganda departments directly monitor WeChat users. In 2019, Protestors in Hong Kong destroyed smart lampposts amid concerns they could contain cameras and facial recognition system used for surveillance by Chinese authorities. Human rights groups have criticized the Chinese government for using artificial intelligence facial recognition technology in its suppression against Uyghurs, Christians and Falun Gong practitioners. ==== India ==== Even though facial recognition technology (FRT) is not fully accurate, it is being increasingly deployed for identification purposes by the police in India. FRT systems generate a probability match score, or a confidence score between the suspect who is to be identified and the database of identified criminals that is available with the police. The National Automated Facial Recognition System (AFRS) is already being developed by the National Crime Records Bureau (NCRB), a body constituted under the Ministry of Home Affairs. The project seeks to develop and deploy a national database of photographs which would comport with a facial recognition technology system by the central and state security agencies. The Internet Freedom Foundation has flagged concerns regarding the project. The NGO has highlighted that the accuracy of FRT systems are "routinely exaggerated and the real numbers leave much to be desired. The implementation of such faulty FRT systems would lead to high rates of false positives and false negatives in this recognition process." Under the Supreme Court of India's decision in Justice K.S. Puttaswamy vs Union of India (22017 10 SCC 1), any justifiable intrusion by the State into people's right to privacy, which is protected as a fundamental right under Article 21 of the Constitution, must confirm to certain thresholds, namely: legality, necessity, proportionality and procedural safeguards. As per the Internet Freedom Foundation, the National Automated Facial Recognition System (AFRS) proposal fails to meet any of these thresholds, citing "absence of legality," "manifest arbitrariness," and "absence of

safeguards and accountability." While the national level AFRS project is still in the works, police departments in various states in India are already deploying facial recognition technology systems, such as: TSCOP + CCTNS in Telangana, Punjab Artificial Intelligence System (PAIS) in Punjab, Trinetra in Uttar Pradesh, Police Artificial Intelligence System in Uttarakhand, AFRS in Delhi, Automated Multimodal Biometric Identification System (AMBIS) in Maharashtra, FaceTagr in Tamil Nadu. The Crime and Criminal Tracking Network and Systems (CCTNS), which is a Mission Mode Project under the National e-Governance Plan (NeGP), is viewed as a system which would connect police stations across India, and help them "talk" to each other. The project's objective is to digitize all FIR-related information, including FIRs registered, as well as cases investigated, charge sheets filed, and suspects and wanted persons in all police stations. This shall constitute a national database of crime and criminals in India. CCTNS is being implemented without a data protection law in place. CCTNS is proposed to be integrated with the AFRS, a repository of all crime and criminal related facial data which can be deployed to purportedly identify or verify a person from a variety of inputs ranging from images to videos. This has raised privacy concerns from civil society organizations and privacy experts. Both the projects have been censured as instruments of "mass surveillance" at the hands of the state. In Rajasthan, 'RajCop,' a police app has been recently integrated with a facial recognition module which can match the face of a suspect against a database of known persons in real-time. Rajasthan police is in currently working to widen the ambit of this module by making it mandatory to upload photographs of all arrested persons in CCTNS database, which will "help develop a rich database of known offenders." Helmets fixed with camera have been designed and being used by Rajasthan police in law and order situations to capture police action and activities of "the miscreants, which can later serve as evidence during the investigation of such cases." PAIS (Punjab Artificial Intelligence System), App employs deep learning, machine learning, and face recognition for the identification of criminals to assist police personnel. The state of Telangana has installed 8 lakh CCTV cameras, with its capital city Hyderabad slowly turning into a surveillance capital. A false positive happens when facial recognition technology misidentifies a person to be someone they are not, that is, it yields an incorrect positive result. They often results in discrimination and strengthening of existing biases. For example, in 2018, Delhi Police reported that its FRT system had an accuracy rate of 2%, which sank to 1% in 2019. The FRT system even failed to distinguish accurately between different sexes. The government of Delhi in collaboration with Indian Space Research Organisation (ISRO) is developing a new technology called Crime Mapping Analytics and Predictive System (CMAPS). The project aims to deploy space technology for "controlling crime and maintaining law and order." The system will be connected to a database containing data of criminals. The technology is envisaged to be deployed to collect real-time data at the crime scene. In a reply dated November 25, 2020 to a Right to Information request filed by the Internet Freedom Foundation seeking information about the facial recognition system being used by the Delhi Police (with reference number DEPOL/R/E/20/07128), the Office of the Deputy Commissioner of Police cum Public Information Officer: Crime stated that they cannot provide the information under section 8(d) of the Right to Information Act, 2005. A Right to Information (RTI) request dated July 30, 2020 was filed with the Office of the Commissioner, Kolkata Police, seeking information about the facial recognition technology that the department was using. The information sought was denied stating that the department was exempted from disclosure under section 24(4) of the RTI Act. ==== Latin America ==== In the 2000 Mexican presidential election, the Mexican government employed face recognition software to prevent voter fraud. Some individuals had been registering to vote under

several different names, in an attempt to place multiple votes. By comparing new face images to those already in the voter database, authorities were able to reduce duplicate registrations. In Colombia public transport busses are fitted with a facial recognition system by FaceFirst Inc to identify passengers that are sought by the National Police of Colombia. FaceFirst Inc also built the facial recognition system for Tocumen International Airport in Panama. The face recognition system is deployed to identify individuals among the travellers that are sought by the Panamanian National Police or Interpol. Tocumen International Airport operates an airport-wide surveillance system using hundreds of live face recognition cameras to identify wanted individuals passing through the airport. The face recognition system was initially installed as part of a US$11 million contract and included a computer cluster of sixty computers, a fiber-optic cable network for the airport buildings, as well as the installation of 150 surveillance cameras in the airport terminal and at about 30 airport gates. At the 2014 FIFA World Cup in Brazil the Federal Police of Brazil used face recognition goggles. Face recognition systems "made in China" were also deployed at the 2016 Summer Olympics in Rio de Janeiro. Nuctech Company provided 145 inspection terminals for Maracanã Stadium and 55 terminals for the Deodoro Olympic Park. ==== European Union ==== Police forces in at least 21 countries of the European Union use, or plan to use, facial recognition systems, either for administrative or criminal purposes. ===== Greece ===== Greek police passed a contract with Intracom-Telecom for the provision of at least 1,000 devices equipped with live facial recognition system. The delivery is expected before the summer 2021. The total value of the contract is over 4 million euros, paid for in large part by the Internal Security Fund of the European Commission. ===== Italy ===== Italian police acquired a face recognition system in 2017, Sistema Automatico Riconoscimento Immagini (SARI). In November 2020, the Interior ministry announced plans to use it in real-time to identify people suspected of seeking asylum. ===== The Netherlands ===== The Netherlands has deployed facial recognition and artificial intelligence technology since 2016. The database of the Dutch police currently contains over 2.2 million pictures of 1.3 million Dutch citizens. This accounts for about 8% of the population. In The Netherlands, face recognition is not used by the police on municipal CCTV. ==== South Africa ==== In South Africa, in 2016, the city of Johannesburg announced it was rolling out smart CCTV cameras complete with automatic number plate recognition and facial recognition. === Deployment in retail stores === The US firm 3VR, now Identiv, is an example of a vendor which began offering facial recognition systems and services to retailers as early as 2007. In 2012, the company advertised benefits such as "dwell and queue line analytics to decrease customer wait times", "facial surveillance analytic[s] to facilitate personalized customer greetings by employees" and the ability to "[c]reate loyalty programs by combining Point of sale (POS) data with facial recognition". ==== United States ==== In 2018, the National Retail Federation Loss Prevention Research Council called facial recognition technology "a promising new tool" worth evaluating. In July 2020, the Reuters news agency reported that during the 2010s the pharmacy chain Rite Aid had deployed facial recognition video surveillance systems and components from FaceFirst, DeepCam LLC, and other vendors at some retail locations in the United States. Cathy Langley, Rite Aid's vice president of asset protection, used the phrase "feature matching" to refer to the systems and said that usage of the systems resulted in less violence and organized crime in the company's stores, while former vice president of asset protection Bob Oberosler emphasized improved safety for staff and a reduced need for the involvement of law enforcement organizations. In a 2020 statement to Reuters in response to the reporting, Rite Aid said that it had ceased using the facial recognition software and switched off the cameras. According to director Read Hayes of the National Retail Federation Loss Prevention Research

Council, Rite Aid's surveillance program was either the largest or one of the largest programs in retail. The Home Depot, Menards, Walmart, and 7-Eleven are among other US retailers also engaged in large-scale pilot programs or deployments of facial recognition technology. Of the Rite Aid stores examined by Reuters in 2020, those in communities where people of color made up the largest racial or ethnic group were three times as likely to have the technology installed, raising concerns related to the substantial history of racial segregation and racial profiling in the United States. Rite Aid said that the selection of locations was "data-driven", based on the theft histories of individual stores, local and national crime data, and site infrastructure. ==== Australia ==== In 2019, facial recognition to prevent theft was in use at Sydney's Star Casino and was also deployed at gaming venues in New Zealand. In June 2022, consumer group CHOICE reported facial recognition was in use in Australia at Kmart, Bunnings, and The Good Guys. The Good Guys subsequently suspended the technology pending a legal challenge by CHOICE to the Office of the Australian Information Commissioner, while Bunnings kept the technology in use and Kmart maintained its trial of the technology. === Additional uses === At the American football championship game Super Bowl XXXV in January 2001, police in Tampa Bay, Florida used Viisage face recognition software to search for potential criminals and terrorists in attendance at the event. 19 people with minor criminal records were potentially identified. Face recognition systems have also been used by photo management software to identify the subjects of photographs, enabling features such as searching images by person, as well as suggesting photos to be shared with a specific contact if their presence were detected in a photo. By 2008 facial recognition systems were typically used as access control in security systems. The United States' popular music and country music celebrity Taylor Swift surreptitiously employed facial recognition technology at a concert in 2018. The camera was embedded in a kiosk near a ticket booth and scanned concert-goers as they entered the facility for known stalkers. On August 18, 2019, The Times reported that the UAE-owned Manchester City hired a Texas-based firm, Blink Identity, to deploy facial recognition systems in a driver program. The club has planned a single super-fast lane for the supporters at the Etihad stadium. However, civil rights groups cautioned the club against the introduction of this technology, saying that it would risk "normalising a mass surveillance tool". The policy and campaigns officer at Liberty, Hannah Couchman said that Man City's move is alarming, since the fans will be obliged to share deeply sensitive personal information with a private company, where they could be tracked and monitored in their everyday lives. In 2019, casinos in Australia and New Zealand rolled out facial recognition to prevent theft, and a representative of Sydney's Star Casino said they would also provide 'customer service' like welcoming a patron back to a bar. In August 2020, amid the COVID-19 pandemic in the United States, American football stadiums of New York and Los Angeles announced the installation of facial recognition for upcoming matches. The purpose is to make the entry process as touchless as possible. Disney's Magic Kingdom, near Orlando, Florida, likewise announced a test of facial recognition technology to create a touchless experience during the pandemic; the test was originally slated to take place between March 23 and April 23, 2021, but the limited timeframe had been removed as of late April 2021. Media companies have begun using face recognition technology to streamline their tracking, organizing, and archiving pictures and videos. == Advantages and disadvantages == === Compared to other biometric systems === In 2006, the performance of the latest face recognition algorithms was evaluated in the Face Recognition Grand Challenge (FRGC). High-resolution face images, 3-D face scans, and iris images were used in the tests. The results indicated that the new algorithms are 10 times more accurate than the face recognition algorithms of 2002 and 100 times more accurate than those of

1995. Some of the algorithms were able to outperform human participants in recognizing faces and could uniquely identify identical twins. One key advantage of a facial recognition system that it is able to perform mass identification as it does not require the cooperation of the test subject to work. Properly designed systems installed in airports, multiplexes, and other public places can identify individuals among the crowd, without passers-by even being aware of the system. However, as compared to other biometric techniques, face recognition may not be most reliable and efficient. Quality measures are very important in facial recognition systems as large degrees of variations are possible in face images. Factors such as illumination, expression, pose and noise during face capture can affect the performance of facial recognition systems. Among all biometric systems, facial recognition has the highest false acceptance and rejection rates, thus questions have been raised on the effectiveness of or bias of face recognition software in cases of railway and airport security, law enforcement and housing and employment decisions. === Weaknesses === Ralph Gross, a researcher at the Carnegie Mellon Robotics Institute in 2008, describes one obstacle related to the viewing angle of the face: "Face recognition has been getting pretty good at full frontal faces and 20 degrees off, but as soon as you go towards profile, there've been problems." Besides the pose variations, low-resolution face images are also very hard to recognize. This is one of the main obstacles of face recognition in surveillance systems. It has also been suggested that camera settings can favour sharper imagery of white skin than of other skin tones. Face recognition is less effective if facial expressions vary. A big smile can render the system less effective. For instance: Canada, in 2009, allowed only neutral facial expressions in passport photos. There is also inconstancy in the datasets used by researchers. Researchers may use anywhere from several subjects to scores of subjects and a few hundred images to thousands of images. Data sets may be diverse and inclusive or mainly contain images of white males. It is important for researchers to make available the datasets they used to each other, or have at least a standard or representative dataset. Although high degrees of accuracy have been claimed for some facial recognition systems, these outcomes are not universal. The consistently worst accuracy rate is for those who are 18 to 30 years old, Black and female. === Racial bias and skin tone === Studies have shown that facial recognition algorithms tend to perform better on individuals with lighter skin tones compared to those with darker skin tones. For example, a 2018 study found that leading commercial gender classification models, which are facial recognition models, have an error rate up to 7 times higher for those with darker skin tones compared to those with lighter skin tones. Initially this disparity was attributed to the imbalance in training datasets, which often overrepresent lighter-skinned individuals, leading to higher error rates for darker-skinned people. However, later studies showed that dataset imbalance is not the sole cause for difference in performance. Common image compression methods, such as JPEG chroma subsampling, have been found to disproportionately degrade performance for darker-skinned individuals. These methods inadequately represent color information, which adversely affects the ability of algorithms to recognize darker-skinned individuals accurately. === Cross-race effect bias === Facial recognition systems often demonstrate lower accuracy when identifying individuals with non-Eurocentric facial features. Known as the Cross-race effect, this bias occurs when systems perform better on racial or ethnic groups that are overrepresented in their training data, resulting in reduced accuracy for underrepresented groups. The overrepresented group is generally the more populous group in the location that the model is being developed. For example, models developed in Asian cultures generally perform better on Asian facial features than Eurocentric facial features due to overrepresentation in the developers training dataset. The opposite is observed in models developed in Eurocentric cultures. The

systems used for facial recognition often lack the sufficient training needed to fully recognize those features not of Eurocentric descent. When the training and databases for these Machine Learning (ML) models do not contain a diverse representation, the models fail to identify the missed population, adding to their racial biases. The cross-race effect is not exclusive to machines; humans also experience difficulty recognizing faces from racial or ethnic groups different from their own. This is an example of inherent human biases being perpetuated in training datasets. === Challenges for individuals with disabilities === Facial recognition technologies encounter significant challenges when identifying individuals with disabilities. For instance, systems have been shown to perform worse when recognizing individuals with Down syndrome, often leading to increased false match rates. This is due to distinct facial structures associated with the condition that are not adequately represented in training datasets. More broadly, facial recognition systems tend to overlook diverse physical characteristics related to disabilities. The lack of representative data for individuals with varying disabilities further emphasizes the need for inclusive algorithmic designs to mitigate bias and improve accuracy. Additionally, facial expression recognition technologies often fail to accurately interpret the emotional states of individuals with intellectual disabilities. This shortcoming can hinder effective communication and interaction, underscoring the necessity for systems trained on diverse datasets that include individuals with intellectual disabilities. Furthermore, biases in facial recognition algorithms can lead to discriminatory outcomes for people with disabilities. For example, certain facial features or asymmetries may result in misidentification or exclusion, highlighting the importance of developing accessible and fair biometric systems. === Advancements in fairness and mitigation strategies === A significant efforts have been made to address the demographic biases in face recognition: by processing at input, model, or output stages. Additionally, targeted dataset collection has been shown to improve racial equity in facial recognition systems. By prioritizing diverse data inputs, researchers demonstrated measurable reductions in performance disparities between racial groups. === Ineffectiveness === Critics of the technology complain that the London Borough of Newham scheme has, as of 2004, never recognized a single criminal, despite several criminals in the system's database living in the Borough and the system has been running for several years. "Not once, as far as the police know, has Newham's automatic face recognition system spotted a live target." This information seems to conflict with claims that the system was credited with a 34% reduction in crime (hence why it was rolled out to Birmingham also). An experiment in 2002 by the local police department in Tampa, Florida, had similarly disappointing results. A system at Boston's Logan Airport was shut down in 2003 after failing to make any matches during a two-year test period. In 2014, Facebook stated that in a standardized two-option facial recognition test, its online system scored 97.25% accuracy, compared to the human benchmark of 97.5%. Systems are often advertised as having accuracy near 100%; this is misleading as the outcomes are not universal. The studies often use samples that are smaller and less diverse than would be necessary for large scale applications. Because facial recognition is not completely accurate, it creates a list of potential matches. A human operator must then look through these potential matches and studies show the operators pick the correct match out of the list only about half the time. This causes the issue of targeting the wrong suspect. == Controversies == === Privacy violations === Civil rights organizations and privacy campaigners such as the Electronic Frontier Foundation, Big Brother Watch and the ACLU express concern that privacy is being compromised by the use of surveillance technologies. Face recognition can be used not just to identify an individual, but also to unearth other personal data associated with an individual – such as other photos featuring the individual, blog posts, social media profiles, Internet

behavior, and travel patterns. Concerns have been raised over who would have access to the knowledge of one's whereabouts and people with them at any given time. Moreover, individuals have limited ability to avoid or thwart face recognition tracking unless they hide their faces. This fundamentally changes the dynamic of day-to-day privacy by enabling any marketer, government agency, or random stranger to secretly collect the identities and associated personal information of any individual captured by the face recognition system. Consumers may not understand or be aware of what their data is being used for, which denies them the ability to consent to how their personal information gets shared. In July 2015, the United States Government Accountability Office conducted a Report to the Ranking Member, Subcommittee on Privacy, Technology and the Law, Committee on the Judiciary, U.S. Senate. The report discussed facial recognition technology's commercial uses, privacy issues, and the applicable federal law. It states that previously, issues concerning facial recognition technology were discussed and represent the need for updating the privacy laws of the United States so that federal law continually matches the impact of advanced technologies. The report noted that some industry, government, and private organizations were in the process of developing, or have developed, "voluntary privacy guidelines". These guidelines varied between the stakeholders, but their overall aim was to gain consent and inform citizens of the intended use of facial recognition technology. According to the report the voluntary privacy guidelines helped to counteract the privacy concerns that arise when citizens are unaware of how their personal data gets put to use. In 2016, Russian company NtechLab caused a privacy scandal in the international media when it launched the FindFace face recognition system with the promise that Russian users could take photos of strangers in the street and link them to a social media profile on the social media platform Vkontakte (VK). In December 2017, Facebook rolled out a new feature that notifies a user when someone uploads a photo that includes what Facebook thinks is their face, even if they are not tagged. Facebook has attempted to frame the new functionality in a positive light, amidst prior backlashes. Facebook's head of privacy, Rob Sherman, addressed this new feature as one that gives people more control over their photos online. "We've thought about this as a really empowering feature," he says. "There may be photos that exist that you don't know about." Facebook's DeepFace has become the subject of several class action lawsuits under the Biometric Information Privacy Act, with claims alleging that Facebook is collecting and storing face recognition data of its users without obtaining informed consent, in direct violation of the 2008 Biometric Information Privacy Act (BIPA). The most recent case was dismissed in January 2016 because the court lacked jurisdiction. In the US, surveillance companies such as Clearview AI are relying on the First Amendment to the United States Constitution to data scrape user accounts on social media platforms for data that can be used in the development of facial recognition systems. In 2019, the Financial Times first reported that facial recognition software was in use in the King's Cross area of London. The development around London's King's Cross mainline station includes shops, offices, Google's UK HQ and part of St Martin's College. According to the UK Information Commissioner's Office: "Scanning people's faces as they lawfully go about their daily lives, in order to identify them, is a potential threat to privacy that should concern us all." The UK Information Commissioner Elizabeth Denham launched an investigation into the use of the King's Cross facial recognition system, operated by the company Argent. In September 2019 it was announced by Argent that facial recognition software would no longer be used at King's Cross. Argent claimed that the software had been deployed between May 2016 and March 2018 on two cameras covering a pedestrian street running through the centre of the development. In October 2019, a report by the deputy London mayor Sophie Linden revealed that in a secret deal the Metropolitan Police had

passed photos of seven people to Argent for use in their King's cross facial recognition system. Automated Facial Recognition was trialled by the South Wales Police on multiple occasions between 2017 and 2019. The use of the technology was challenged in court by a private individual, Edward Bridges, with support from the charity Liberty (case known as R (Bridges) v Chief Constable South Wales Police). The case was heard in the Court of Appeal and a judgement was given in August 2020. The case argued that the use of Facial Recognition was a privacy violation on the basis that there was insufficient legal framework or proportionality in the use of Facial Recognition and that its use was in violation of the Data Protection Acts 1998 and 2018. The case was decided in favour of Bridges and did not award damages. The case was settled via a declaration of wrongdoing. In response to the case, the British Government has repeatedly attempted to pass a Bill regulating the use of Facial Recognition in public spaces. The proposed Bills have attempted to appoint a Commissioner with the ability to regulate Facial Recognition use by Government Services in a similar manner to the Commissioner for CCTV. Such a Bill has yet to come into force [correct as of September 2021]. In January 2023, New York Attorney General Letitia James asked for more information on the use of facial recognition technology from Madison Square Garden Entertainment following reports that the firm used it to block lawyers involved in litigation against the company from entering Madison Square Garden. She noted such a move would could go against federal, state, and local human rights laws. === Imperfect technology in law enforcement === As of 2018, it is still contested as to whether or not facial recognition technology works less accurately on people of color. One study by Joy Buolamwini (MIT Media Lab) and Timnit Gebru (Microsoft Research) found that the error rate for gender recognition for women of color within three commercial facial recognition systems ranged from 23.8% to 36%, whereas for lighter-skinned men it was between 0.0 and 1.6%. Overall accuracy rates for identifying men (91.9%) were higher than for women (79.4%), and none of the systems accommodated a non-binary understanding of gender. It also showed that the datasets used to train commercial facial recognition models were unrepresentative of the broader population and skewed toward lighter-skinned males. However, another study showed that several commercial facial recognition software sold to law enforcement offices around the country had a lower false non-match rate for black people than for white people. Experts fear that face recognition systems may actually be hurting citizens the police claims they are trying to protect. It is considered an imperfect biometric, and in a study conducted by Georgetown University researcher Clare Garvie, she concluded that "there's no consensus in the scientific community that it provides a positive identification of somebody." It is believed that with such large margins of error in this technology, both legal advocates and facial recognition software companies say that the technology should only supply a portion of the case – no evidence that can lead to an arrest of an individual. The lack of regulations holding facial recognition technology companies to requirements of racially biased testing can be a significant flaw in the adoption of use in law enforcement. CyberExtruder, a company that markets itself to law enforcement said that they had not performed testing or research on bias in their software. CyberExtruder did note that some skin colors are more difficult for the software to recognize with current limitations of the technology. "Just as individuals with very dark skin are hard to identify with high significance via facial recognition, individuals with very pale skin are the same," said Blake Senftner, a senior software engineer at CyberExtruder. The United States' National Institute of Standards and Technology (NIST) carried out extensive testing of FRT system 1:1 verification and 1:many identification. It also tested for the differing accuracy of FRT across different demographic groups. The independent study concluded at present, no FRT system has 100%

accuracy. === Data protection === In 2010, Peru passed the Law for Personal Data Protection, which defines biometric information that can be used to identify an individual as sensitive data. In 2012, Colombia passed a comprehensive Data Protection Law which defines biometric data as senstivite information. According to Article 9(1) of the EU's 2016 General Data Protection Regulation (GDPR) the processing of biometric data for the purpose of "uniquely identifying a natural person" is sensitive and the facial recognition data processed in this way becomes sensitive personal data. In response to the GDPR passing into the law of EU member states, EU based researchers voiced concern that if they were required under the GDPR to obtain individual's consent for the processing of their facial recognition data, a face database on the scale of MegaFace could never be established again. In September 2019 the Swedish Data Protection Authority (DPA) issued its first ever financial penalty for a violation of the EU's General Data Protection Regulation (GDPR) against a school that was using the technology to replace time-consuming roll calls during class. The DPA found that the school illegally obtained the biometric data of its students without completing an impact assessment. In addition the school did not make the DPA aware of the pilot scheme. A 200,000 SEK fine (€19,000/$21,000) was issued. In the United States of America several U.S. states have passed laws to protect the privacy of biometric data. Examples include the Illinois Biometric Information Privacy Act (BIPA) and the California Consumer Privacy Act (CCPA). In March 2020 California residents filed a class action against Clearview AI, alleging that the company had illegally collected biometric data online and with the help of face recognition technology built up a database of biometric data which was sold to companies and police forces. At the time Clearview AI already faced two lawsuits under BIPA and an investigation by the Privacy Commissioner of Canada for compliance with the Personal Information Protection and Electronic Documents Act (PIPEDA). == Bans on the use of facial recognition technology == === United States of America === In May 2019, San Francisco, California became the first major United States city to ban the use of facial recognition software for police and other local government agencies' usage. San Francisco Supervisor, Aaron Peskin, introduced regulations that will require agencies to gain approval from the San Francisco Board of Supervisors to purchase surveillance technology. The regulations also require that agencies publicly disclose the intended use for new surveillance technology. In June 2019, Somerville, Massachusetts became the first city on the East Coast to ban face surveillance software for government use, specifically in police investigations and municipal surveillance. In July 2019, Oakland, California banned the usage of facial recognition technology by city departments. The American Civil Liberties Union ("ACLU") has campaigned across the United States for transparency in surveillance technology and has supported both San Francisco and Somerville's ban on facial recognition software. The ACLU works to challenge the secrecy and surveillance with this technology. During the George Floyd protests, use of facial recognition by city government was banned in Boston, Massachusetts. As of June 10, 2020, municipal use has been banned in: Berkeley, California Oakland, California Boston, Massachusetts – June 30, 2020 Brookline, Massachusetts Cambridge, Massachusetts Northampton, Massachusetts Springfield, Massachusetts Somerville, Massachusetts Portland, Oregon – September 2020 The West Lafayette, Indiana City Council passed an ordinance banning facial recognition surveillance technology. On October 27, 2020, 22 human rights groups called upon the University of Miami to ban facial recognition technology. This came after the students accused the school of using the software to identify student protesters. The allegations were, however, denied by the university. A state police reform law in Massachusetts will take effect in July 2021; a ban passed by the

legislature was rejected by governor Charlie Baker. Instead, the law requires a judicial warrant, limit the personnel who can perform the search, record data about how the technology is used, and create a commission to make recommendations about future regulations. Reports in 2024 revealed that some police departments, including San Francisco Police Department, had skirted bans on facial recognition technology that had been enacted in their respective cities. === European Union === In January 2020, the European Union suggested, but then quickly scrapped, a proposed moratorium on facial recognition in public spaces. The European "Reclaim Your Face" coalition launched in October 2020. The coalition calls for a ban on facial recognition and launched a European Citizens' Initiative in February 2021. More than 60 organizations call on the European Commission to strictly regulate the use of biometric surveillance technologies. == Emotion recognition == In the 18th and 19th century, the belief that facial expressions revealed the moral worth or true inner state of a human was widespread and physiognomy was a respected science in the Western world. From the early 19th century onwards photography was used in the physiognomic analysis of facial features and facial expression to detect insanity and dementia. In the 1960s and 1970s the study of human emotions and its expressions was reinvented by psychologists, who tried to define a normal range of emotional responses to events. The research on automated emotion recognition has since the 1970s focused on facial expressions and speech, which are regarded as the two most important ways in which humans communicate emotions to other humans. In the 1970s the Facial Action Coding System (FACS) categorization for the physical expression of emotions was established. Its developer Paul Ekman maintains that there are six emotions that are universal to all human beings and that these can be coded in facial expressions. Research into automatic emotion specific expression recognition has in the past decades focused on frontal view images of human faces. Facial thermography can be considered as a promising tool of emotion recognition. In 2016, facial feature emotion recognition algorithms were among the new technologies, alongside high-definition CCTV, high resolution 3D face recognition and iris recognition, that found their way out of university research labs. In 2016, Facebook acquired FacioMetrics, a facial feature emotion recognition corporate spin-off by Carnegie Mellon University. In the same year Apple Inc. acquired the facial feature emotion recognition start-up Emotient. By the end of 2016, commercial vendors of facial recognition systems offered to integrate and deploy emotion recognition algorithms for facial features. The MIT's Media Lab spin-off Affectiva by late 2019 offered a facial expression emotion detection product that can recognize emotions in humans while driving. == Anti-facial recognition systems == The development of anti-facial recognition technology is effectively an arms race between privacy researchers and big data companies. Big data companies increasingly use convolutional AI technology to create ever more advanced facial recognition models. Solutions to block facial recognition may not work on newer software, or on different types of facial recognition models. One popular cited example of facial-recognition blocking is the CVDazzle makeup and haircut system, but the creators note on their website that it has been outdated for quite some time as it was designed to combat a particular facial recognition algorithm and may not work. Another example is the emergence of facial recognition that can identify people wearing facemasks and sunglasses, especially after the COVID-19 pandemic. Given that big data companies have much more funding than privacy researchers, it is very difficult for anti-facial recognition systems to keep up. There is also no guarantee that obfuscation techniques that were used for images taken in the past and stored, such as masks or software obfuscation, would protect users from facial-recognition analysis of those images by future technology. In January 2013, Japanese researchers from the National Institute of Informatics created 'privacy visor' glasses that

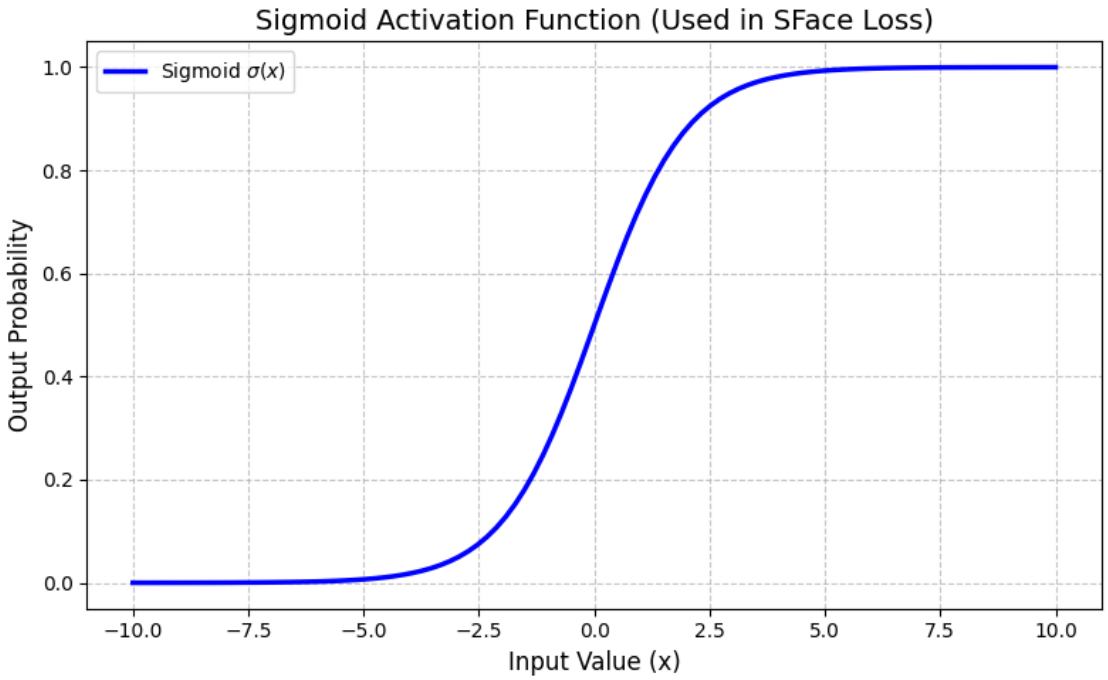
use nearly infrared light to make the face underneath it unrecognizable to face recognition software that use infrared. The latest version uses a titanium frame, light-reflective material and a mask which uses angles and patterns to disrupt facial recognition technology through both absorbing and bouncing back light sources. However, these methods are used to prevent infrared facial recognition and would not work on AI facial recognition of plain images. Some projects use adversarial machine learning to come up with new printed patterns that confuse existing face recognition software. One method that may work to protect from facial recognition systems are specific haircuts and make-up patterns that prevent the used algorithms to detect a face, known as computer vision dazzle. Incidentally, the makeup styles popular with Juggalos may also protect against facial recognition. Facial masks that are worn to protect from contagious viruses can reduce the accuracy of facial recognition systems. A 2020 NIST study, tested popular one-to-one matching systems and found a failure rate between five and fifty percent on masked individuals. The Verge speculated that the accuracy rate of mass surveillance systems, which were not included in the study, would be even less accurate than the accuracy of one-to-one matching systems. The facial recognition of Apple Pay can work through many barriers, including heavy makeup, thick beards and even sunglasses, but fails with masks. However, facial recognition of masked faces is increasingly getting more reliable. Another solution is the application of obfuscation to images that may fool facial recognition systems while still appearing normal to a human user. These could be used for when images are posted online or on social media. However, as it is hard to remove images once they are on the internet, the obfuscation on these images may be defeated and the face of the user identified by future advances in technology. Two examples of this technique, developed in 2020, are the ANU's 'Camera Adversaria' camera app, and the University of Chicago's Fawkes image cloaking software algorithm which applies obfuscation to already taken photos. However, by 2021 the Fawkes obfuscation algorithm had already been specifically targeted by Microsoft Azure which changed its algorithm to lower Fawkes' effectiveness. == See also == Lists List of computer vision topics List of emerging technologiesOutline of artificial intelligence == References == == Further reading == Farokhi, Sajad; Shamsuddin, Siti Mariyam; Flusser, Jan; Sheikh, U.U; Khansari, Mohammad; Jafari-Khouzani, Kourosh (2014). "Near infrared face recognition by combining Zernike moments and undecimated discrete wavelet transform". Digital Signal Processing. 31 (1): 13–27. Bibcode:2014DSP....31...13F. doi:10.1016/j.dsp.2014.04.008. "The Face Detection Algorithm Set to Revolutionize Image Search" (Feb. 2015), MIT Technology Review Garvie, Clare; Bedoya, Alvaro; Frankle, Jonathan (October 18, 2016). Perpetual Line Up: Unregulated Police Face Recognition in America. Center on Privacy & Technology at Georgetown Law. Retrieved October 22, 2016. "Facial Recognition Software 'Sounds Like Science Fiction,' but May Affect Half of Americans". As It Happens. Canadian Broadcasting Corporation. October 20, 2016. Retrieved October 22, 2016. Interview with Alvaro Bedoya, executive director of the Center on Privacy & Technology at Georgetown Law and co-author of Perpetual Line Up: Unregulated Police Face Recognition in America. Press, Eyal, "In Front of Their Faces: Does facial-recognition technology lead police to ignore contradictory evidence?", The New Yorker, 20 November 2023, pp. 20–26. == External links == Media related to Facial recognition system at Wikimedia Commons A Photometric Stereo Approach to Face Recognition (master's thesis). The University of the West of England, Bristol.

3.2 Identity Embeddings

SFace (Sigmoid-Constrained Hypersphere Loss Face Recognition) maps a face image into a 128-dimensional continuous vector space. In this hypersphere, images of the same person are clustered tightly together, while images of different people are pushed far apart.

3.3 The Sigmoid Loss Function

Previous models like ArcFace used fixed margins to force clusters apart. However, noisy datasets (mislabeled or blurry images) would cause the model to over-correct. SFace introduces a Sigmoid-constrained loss that scales gradients based on difficulty. 'Hard' samples are ignored if they are too noisy, preventing the model from corrupting the feature space.



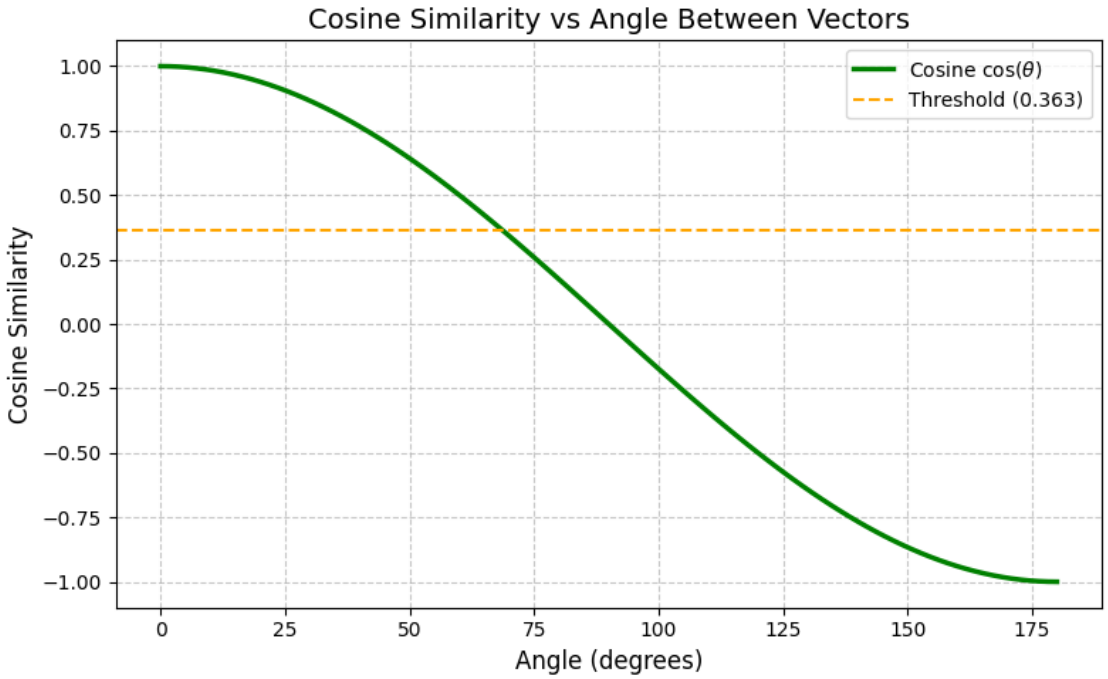
3.4 Vector Matching via Cosine Similarity

To compare two 128-dimensional vectors (A and B), SFace calculates the Cosine Similarity.

$$\textbf{\textit{Cosine}}(\theta) = (A \bullet B) / (||A|| * ||B||)$$

Because SFace normalizes vectors to a length of 1 (hypersphere projection), ||A|| and ||B|| equal 1, reducing the math to a simple dot product:

$$\textbf{\textit{Similarity}} = \Sigma\ (A\_i * B\_i)$$



In this project, a threshold of 0.363 is used. A score above 0.363 indicates the vectors are separated by a small enough angle to be considered the same identity.

Part IV: Liveness and Anti-Spoofing Geometry

4.1 Presentation Attack Detection

Biometrics are body measurements and calculations related to human characteristics and features. Biometric authentication (or realistic authentication) is used in computer science as a form of identification and access control. It is also used to identify individuals in groups that are under surveillance. Biometric identifiers are the distinctive, measurable characteristics used to label and describe individuals. Biometric identifiers are often categorized as physiological characteristics which are related to the shape of the body. Examples include, but are not limited to fingerprint, palm veins, face recognition, DNA, palm print, hand geometry, iris recognition, retina, odor/scent, voice, shape of ears and gait. Behavioral characteristics are related to the pattern of behavior of a person, including but not limited to mouse movement, typing rhythm, gait, signature, voice, and behavioral profiling. Some researchers have coined the term behaviometrics (behavioral biometrics) to describe the latter class of biometrics. More traditional means of access control include token-based identification systems, such as a driver's license or passport, and knowledge-based identification systems, such as a password or personal identification number. Since biometric identifiers are unique to individuals, they are more reliable in verifying identity than token and knowledge-based methods; however, the collection of biometric identifiers raises privacy concerns. == Biometric functionality == Many different aspects of human physiology, chemistry or behavior can be used for biometric authentication. The selection of a particular biometric for use in a specific application involves a weighting of several factors. Jain et al. (1999) identified seven such factors to be used when assessing the suitability of any trait for use in biometric authentication. Biometric authentication is based upon biometric recognition which is an advanced method of recognizing biological and behavioural characteristics of an Individual. Universality means that every person using a system should possess the trait. Uniqueness means the trait should be sufficiently different for individuals in the relevant population such that they can be distinguished from one another. Permanence relates to the manner in which a trait varies over time. More specifically, a trait with good permanence will be reasonably invariant over time with respect to the specific matching algorithm. Measurability (collectability) relates to the ease of acquisition or measurement of the trait. In addition, acquired data should be in a form that permits subsequent processing and extraction of the relevant feature sets. Performance relates to the accuracy, speed, and robustness of technology used (see performance section for more details). Acceptability relates to how well individuals in the relevant population accept the technology such that they are willing to have their biometric trait captured and assessed. Circumvention relates to the ease with which a trait might be imitated using an artifact or substitute. Proper biometric use is very application dependent. Certain biometrics will be better than others based on the required levels of convenience and security. No single biometric will meet all the requirements of every possible application. The block diagram illustrates the two basic modes of a biometric system. First, in verification (or authentication) mode the system performs a one-to-one comparison of a captured biometric with a specific template stored in a biometric database in order to verify the individual is the person they claim to be. Three steps are involved in the verification of a person. In the first step, reference models for all the users are generated and stored in the model database. In the second step, some samples are matched

with reference models to generate the genuine and impostor scores and calculate the threshold. The third step is the testing step. This process may use a smart card, username, or ID number (e.g. PIN) to indicate which template should be used for comparison. Positive recognition is a common use of the verification mode, "where the aim is to prevent multiple people from using the same identity". Second, in identification mode the system performs a one-to-many comparison against a biometric database in an attempt to establish the identity of an unknown individual. The system will succeed in identifying the individual if the comparison of the biometric sample to a template in the database falls within a previously set threshold. Identification mode can be used either for positive recognition (so that the user does not have to provide any information about the template to be used) or for negative recognition of the person "where the system establishes whether the person is who she (implicitly or explicitly) denies to be". The latter function can only be achieved through biometrics since other methods of personal recognition, such as passwords, PINs, or keys, are ineffective. The first time an individual uses a biometric system is called enrollment. During enrollment, biometric information from an individual is captured and stored. In subsequent uses, biometric information is detected and compared with the information stored at the time of enrollment. Note that it is crucial that storage and retrieval of such systems themselves be secure if the biometric system is to be robust. The first block (sensor) is the interface between the real world and the system; it has to acquire all the necessary data. Most of the time it is an image acquisition system, but it can change according to the characteristics desired. The second block performs all the necessary pre-processing: it has to remove artifacts from the sensor, to enhance the input (e.g. removing background noise), to use some kind of normalization, etc. In the third block, necessary features are extracted. This step is an important step as the correct features need to be extracted in an optimal way. A vector of numbers or an image with particular properties is used to create a template. A template is a synthesis of the relevant characteristics extracted from the source. Elements of the biometric measurement that are not used in the comparison algorithm are discarded in the template to reduce the file size and to protect the identity of the enrollee. However, depending on the scope of the biometric system, original biometric image sources may be retained, such as the PIV-cards used in the Federal Information Processing Standard Personal Identity Verification (PIV) of Federal Employees and Contractors (FIPS 201). During the enrollment phase, the template is simply stored somewhere (on a card or within a database or both). During the matching phase, the obtained template is passed to a matcher that compares it with other existing templates, estimating the distance between them using any algorithm (e.g. Hamming distance). The matching program will analyze the template with the input. This will then be output for a specified use or purpose (e.g. entrance in a restricted area), though it is a fear that the use of biometric data may face mission creep. Selection of biometrics in any practical application depending upon the characteristic measurements and user requirements. In selecting a particular biometric, factors to consider include, performance, social acceptability, ease of circumvention and/or spoofing, robustness, population coverage, size of equipment needed and identity theft deterrence. The selection of a biometric is based on user requirements and considers sensor and device availability, computational time and reliability, cost, sensor size, and power consumption. == Multimodal biometric system == Multimodal biometric systems use multiple sensors or biometrics to overcome the limitations of unimodal biometric systems. For instance iris recognition systems can be compromised by aging irises and electronic fingerprint recognition can be worsened by worn-out or cut fingerprints. While unimodal biometric systems are limited by the integrity of their identifier, it is unlikely that several unimodal systems will suffer from identical limitations. Multimodal biometric

systems can obtain sets of information from the same marker (i.e., multiple images of an iris, or scans of the same finger) or information from different biometrics (requiring fingerprint scans and, using voice recognition, a spoken passcode). Multimodal biometric systems can fuse these unimodal systems sequentially, simultaneously, a combination thereof, or in series, which refer to sequential, parallel, hierarchical and serial integration modes, respectively. Fusion of the biometrics information can occur at different stages of a recognition system. In case of feature level fusion, the data itself or the features extracted from multiple biometrics are fused. Matching-score level fusion consolidates the scores generated by multiple classifiers pertaining to different modalities. Finally, in case of decision level fusion the final results of multiple classifiers are combined via techniques such as majority voting. Feature level fusion is believed to be more effective than the other levels of fusion because the feature set contains richer information about the input biometric data than the matching score or the output decision of a classifier. Therefore, fusion at the feature level is expected to provide better recognition results. Spoof attacks consist in submitting fake biometric traits to biometric systems, and are a major threat that can curtail their security. Multi-modal biometric systems are commonly believed to be intrinsically more robust to spoof attacks, but recent studies have shown that they can be evaded by spoofing even a single biometric trait. One such proposed system of Multimodal Biometric Cryptosystem Involving the Face, Fingerprint, and Palm Vein by Prasanalakshmi The Cryptosystem Integration combines biometrics with cryptography, where the palm vein acts as a cryptographic key, offering a high level of security since palm veins are unique and difficult to forge. The Fingerprint Involves minutiae extraction (terminations and bifurcations) and matching techniques. Steps include image enhancement, binarization, ROI extraction, and minutiae thinning. The Face system uses class-based scatter matrices to calculate features for recognition, and the Palm Vein acts as an unbreakable cryptographic key, ensuring only the correct user can access the system. The cancelable Biometrics concept allows biometric traits to be altered slightly to ensure privacy and avoid theft. If compromised, new variations of biometric data can be issued. The Encryption fingerprint template is encrypted using the palm vein key via XOR operations. This encrypted Fingerprint is hidden within the face image using steganographic techniques. Enrollment and Verification for the Biometric data (Fingerprint, palm vein, face) are captured, encrypted, and embedded into a face image. The system extracts the biometric data and compares it with stored values for Verification. The system was tested with fingerprint databases, achieving 75% verification accuracy at an equal error rate of 25% and processing time approximately 50 seconds for enrollment and 22 seconds for Verification. High security due to palm vein encryption, effective against biometric spoofing, and the multimodal approach ensures reliability if one biometric fails. Potential for integration with smart cards or on-card systems, enhancing security in personal identification systems. == Performance == The discriminating powers of all biometric technologies depend on the amount of entropy they are able to encode and use in matching. The following are used as performance metrics for biometric systems: False match rate (FMR, also called FAR = False Accept Rate): the probability that the system incorrectly matches the input pattern to a non-matching template in the database. It measures the percent of invalid inputs that are incorrectly accepted. In case of similarity scale, if the person is an imposter in reality, but the matching score is higher than the threshold, then he is treated as genuine. This increases the FMR, which thus also depends upon the threshold value. False non-match rate (FNMR, also called FRR = False Reject Rate): the probability that the system fails to detect a match between the input pattern and a matching template in the database. It measures the percent of valid inputs that are incorrectly rejected. Receiver operating characteristic

or relative operating characteristic (ROC): The ROC plot is a visual characterization of the trade-off between the FMR and the FNMR. In general, the matching algorithm performs a decision based on a threshold that determines how close to a template the input needs to be for it to be considered a match. If the threshold is reduced, there will be fewer false non-matches but more false accepts. Conversely, a higher threshold will reduce the FMR but increase the FNMR. A common variation is the Detection error trade-off (DET), which is obtained using normal deviation scales on both axes. This more linear graph illuminates the differences for higher performances (rarer errors). Equal error rate or crossover error rate (EER or CER): the rate at which both acceptance and rejection errors are equal. The value of the EER can be easily obtained from the ROC curve. The EER is a quick way to compare the accuracy of devices with different ROC curves. In general, the device with the lowest EER is the most accurate. Failure to enroll rate (FTE or FER): the rate at which attempts to create a template from an input is unsuccessful. This is most commonly caused by low-quality inputs. Failure to capture rate (FTC): Within automatic systems, the probability that the system fails to detect a biometric input when presented correctly. Template capacity: the maximum number of sets of data that can be stored in the system. == History == An early cataloguing of fingerprints dates back to 1885 when Juan Vucetich started a collection of fingerprints of criminals in Argentina. Josh Ellenbogen and Nitzan Lebovic argued that Biometrics originated in the identification systems of criminal activity developed by Alphonse Bertillon (1853–1914) and by Francis Galton's theory of fingerprints and physiognomy. Galton's journey to South Africa from 1850-1852 sparked the beginning of the history of biometric government. Historians note that Galton's travels exposed him to the violence of the colonial frontier, which reinforced his early racial prejudices and inspired his later commitment to classifying human difference scientifically. After returning to England, Galton found a receptive audience for these ideas and influenced Charles Darwin toward a more hierarchical interpretation of human evolution, helping to give the phrase "survival of the fittest" its later association with eugenics. According to Lebovic, Galton's work "led to the application of mathematical models to fingerprints, phrenology, and facial characteristics", as part of "absolute identification" and "a key to both inclusion and exclusion" of populations. Accordingly, "the biometric system is the absolute political weapon of our era" and a form of "soft control". The theoretician David Lyon showed that during the past two decades biometric systems have penetrated the civilian market, and blurred the lines between governmental forms of control and private corporate control. Kelly A. Gates identified 9/11 as the turning point for the cultural language of our present: "in the language of cultural studies, the aftermath of 9/11 was a moment of articulation, where objects or events that have no necessary connection come together and a new discourse formation is established: automated facial recognition as a homeland security technology." == Adaptive biometric systems == Adaptive biometric systems aim to auto-update the templates or model to the intra-class variation of the operational data. The two-fold advantages of these systems are solving the problem of limited training data and tracking the temporal variations of the input data through adaptation. Recently, adaptive biometrics have received a significant attention from the research community. This research direction is expected to gain momentum because of their key promulgated advantages. First, with an adaptive biometric system, one no longer needs to collect a large number of biometric samples during the enrollment process. Second, it is no longer necessary to enroll again or retrain the system from scratch in order to cope with the changing environment. This convenience can significantly reduce the cost of maintaining a biometric system. Despite these advantages, there are several open issues involved with these systems. For mis-classification error (false acceptance) by the biometric system, cause adaptation

using impostor sample. However, continuous research efforts are directed to resolve the open issues associated to the field of adaptive biometrics. More information about adaptive biometric systems can be found in the critical review by Rattani et al. == Recent advances in emerging biometrics == In recent times, biometrics based on brain (electroencephalogram) and heart (electrocardiogram) signals have emerged. An example is finger vein recognition, using pattern-recognition techniques, based on images of human vascular patterns. The advantage of this newer technology is that it is more fraud resistant compared to conventional biometrics like fingerprints. However, such technology is generally more cumbersome and still has issues such as lower accuracy and poor reproducibility over time. On the portability side of biometric products, more and more vendors are embracing significantly miniaturized biometric authentication systems (BAS) thereby driving elaborate cost savings, especially for large-scale deployments. === Operator signatures === An operator signature is a biometric mode where the manner in which a person using a device or complex system is recorded as a verification template. One potential use for this type of biometric signature is to distinguish among remote users of telerobotic surgery systems that utilize public networks for communication. === Proposed requirement for certain public networks === John Michael (Mike) McConnell, a former vice admiral in the United States Navy, a former director of U.S. National Intelligence, and senior vice president of Booz Allen Hamilton, promoted the development of a future capability to require biometric authentication to access certain public networks in his keynote speech at the 2009 Biometric Consortium Conference. A basic premise in the above proposal is that the person that has uniquely authenticated themselves using biometrics with the computer is in fact also the agent performing potentially malicious actions from that computer. However, if control of the computer has been subverted, for example in which the computer is part of a botnet controlled by a hacker, then knowledge of the identity of the user at the terminal does not materially improve network security or aid law enforcement activities. === Animal biometrics === Rather than tags or tattoos, biometric techniques may be used to identify individual animals: zebra stripes, blood vessel patterns in rodent ears, muzzle prints, bat wing patterns, primate facial recognition and koala spots have all been tried. == Issues and concerns == === Human dignity === Biometrics have been considered also instrumental to the development of state authority (to put it in Foucauldian terms, of discipline and biopower). By turning the human subject into a collection of biometric parameters, biometrics would dehumanize the person, infringe bodily integrity, and, ultimately, offend human dignity. In a well-known case, Italian philosopher Giorgio Agamben refused to enter the United States in protest at the United States Visitor and Immigrant Status Indicator (US-VISIT) program's requirement for visitors to be fingerprinted and photographed. Agamben argued that gathering of biometric data is a form of bio-political tattooing, akin to the tattooing of Jews during the Holocaust. According to Agamben, biometrics turn the human persona into a bare body. Agamben refers to the two words used by Ancient Greeks for indicating "life", zoe, which is the life common to animals and humans, just life; and bios, which is life in the human context, with meanings and purposes. Agamben envisages the reduction to bare bodies for the whole humanity. For him, a new bio-political relationship between citizens and the state is turning citizens into pure biological life (zoe) depriving them from their humanity (bios); and biometrics would herald this new world. In Dark Matters: On the Surveillance of Blackness, surveillance scholar Simone Browne formulates a similar critique as Agamben, citing a recent study relating to biometrics R&D; that found that the gender classification system being researched "is inclined to classify Africans as males and Mongoloids as females." Consequently, Browne argues that the conception of an objective biometric technology is difficult if such systems are subjectively

designed, and are vulnerable to cause errors as described in the study above. The stark expansion of biometric technologies in both the public and private sector magnifies this concern. The increasing commodification of biometrics by the private sector adds to this danger of loss of human value. Indeed, corporations value the biometric characteristics more than the individuals value them. Browne goes on to suggest that modern society should incorporate a "biometric consciousness" that "entails informed public debate around these technologies and their application, and accountability by the state and the private sector, where the ownership of and access to one's own body data and other intellectual property that is generated from one's body data must be understood as a right." Other scholars have emphasized, however, that the globalized world is confronted with a huge mass of people with weak or absent civil identities. Most developing countries have weak and unreliable documents and the poorer people in these countries do not have even those unreliable documents. Without certified personal identities, there is no certainty of right, no civil liberty. One can claim his rights, including the right to refuse to be identified, only if he is an identifiable subject, if he has a public identity. In such a sense, biometrics could play a pivotal role in supporting and promoting respect for human dignity and fundamental rights. === Privacy and discrimination === It is possible that data obtained during biometric enrollment may be used in ways for which the enrolled individual has not consented. For example, most biometric features could disclose physiological and/or pathological medical conditions (e.g., some fingerprint patterns are related to chromosomal diseases, iris patterns could reveal sex, hand vein patterns could reveal vascular diseases, most behavioral biometrics could reveal neurological diseases, etc.). Moreover, second generation biometrics, notably behavioral and electro-physiologic biometrics (e.g., based on electrocardiography, electroencephalography, electromyography), could be also used for emotion detection. There are three categories of privacy concerns: Unintended functional scope: The authentication goes further than authentication, such as finding a tumor. Unintended application scope: The authentication process correctly identifies the subject when the subject did not wish to be identified. Covert identification: The subject is identified without seeking identification or authentication, i.e. a subject's face is identified in a crowd. In terms of recognition or identification performance for a given trait, a biometric system should not exhibit accuracy differences across demographic groups. Disparities in accuracy can lead to uneven error rates for populations defined by gender, race, age, or other attributes . Therefore, biometric systems should be designed and evaluated to ensure demographic fairness, providing equitable performance for all users. === Danger to owners of secured items === When thieves cannot get access to secure properties, there is a chance that the thieves will stalk and assault the property owner to gain access. If the item is secured with a biometric device, the damage to the owner could be irreversible, and potentially cost more than the secured property. For example, in 2005, Malaysian car thieves cut off a man's finger when attempting to steal his Mercedes-Benz S-Class. === Attacks at presentation === In the context of biometric systems, presentation attacks may also be called "spoofing attacks". As per the recent ISO/IEC 30107 standard, presentation attacks are defined as "presentation to the biometric capture subsystem with the goal of interfering with the operation of the biometric system". These attacks can be either impersonation or obfuscation attacks. Impersonation attacks try to gain access by pretending to be someone else. Obfuscation attacks may, for example, try to evade face detection and face recognition systems. Several methods have been proposed to counteract presentation attacks. === Surveillance humanitarianism in times of crisis === Biometrics are employed by many aid programs in times of crisis in order to prevent fraud and ensure that resources are properly available to those in need. Humanitarian efforts are motivated by promoting

the welfare of individuals in need, however the use of biometrics as a form of surveillance humanitarianism can create conflict due to varying interests of the groups involved in the particular situation. Disputes over the use of biometrics between aid programs and party officials stalls the distribution of resources to people that need help the most. In July 2019, the United Nations World Food Program and Houthi Rebels were involved in a large dispute over the use of biometrics to ensure resources are provided to the hundreds of thousands of civilians in Yemen whose lives are threatened. The refusal to cooperate with the interests of the United Nations World Food Program resulted in the suspension of food aid to the Yemen population. The use of biometrics may provide aid programs with valuable information, however its potential solutions may not be best suited for chaotic times of crisis. Conflicts that are caused by deep-rooted political problems, in which the implementation of biometrics may not provide a long-term solution. === Cancelable biometrics === One advantage of passwords over biometrics is that they can be re-issued. If a token or a password is lost or stolen, it can be cancelled and replaced by a newer version. This is not naturally available in biometrics. If someone's face is compromised from a database, they cannot cancel or reissue it. If the electronic biometric identifier is stolen, it is nearly impossible to change a biometric feature. This renders the person's biometric feature questionable for future use in authentication, such as the case with the hacking of security-clearance-related background information from the Office of Personnel Management (OPM) in the United States. Cancelable biometrics is a way in which to incorporate protection and the replacement features into biometrics to create a more secure system. It was first proposed by Ratha et al. "Cancelable biometrics refers to the intentional and systematically repeatable distortion of biometric features in order to protect sensitive user-specific data. If a cancelable feature is compromised, the distortion characteristics are changed, and the same biometrics is mapped to a new template, which is used subsequently. Cancelable biometrics is one of the major categories for biometric template protection purpose besides biometric cryptosystem." In biometric cryptosystem, "the error-correcting coding techniques are employed to handle intraclass variations." This ensures a high level of security but has limitations such as specific input format of only small intraclass variations. Several methods for generating new exclusive biometrics have been proposed. The first fingerprint-based cancelable biometric system was designed and developed by Tulyakov et al. Essentially, cancelable biometrics perform a distortion of the biometric image or features before matching. The variability in the distortion parameters provides the cancelable nature of the scheme. Some of the proposed techniques operate using their own recognition engines, such as Teoh et al. and Savvides et al., whereas other methods, such as Dabbah et al., take the advantage of the advancement of the well-established biometric research for their recognition front-end to conduct recognition. Although this increases the restrictions on the protection system, it makes the cancellable templates more accessible for available biometric technologies === Proposed soft biometrics === Soft biometrics are understood as not strict biometrical recognition practices that are proposed in favour of identity cheaters and stealers. Traits are physical, behavioral or adhered human characteristics that have been derived from the way human beings normally distinguish their peers (e.g. height, gender, hair color). They are used to complement the identity information provided by the primary biometric identifiers. Although soft biometric characteristics lack the distinctiveness and permanence to recognize an individual uniquely and reliably, and can be easily faked, they provide some evidence about the users identity that could be beneficial. In other words, despite the fact they are unable to individualize a subject, they are effective in distinguishing between people. Combinations of personal attributes like gender, race, eye color, height and other visible identification marks can be

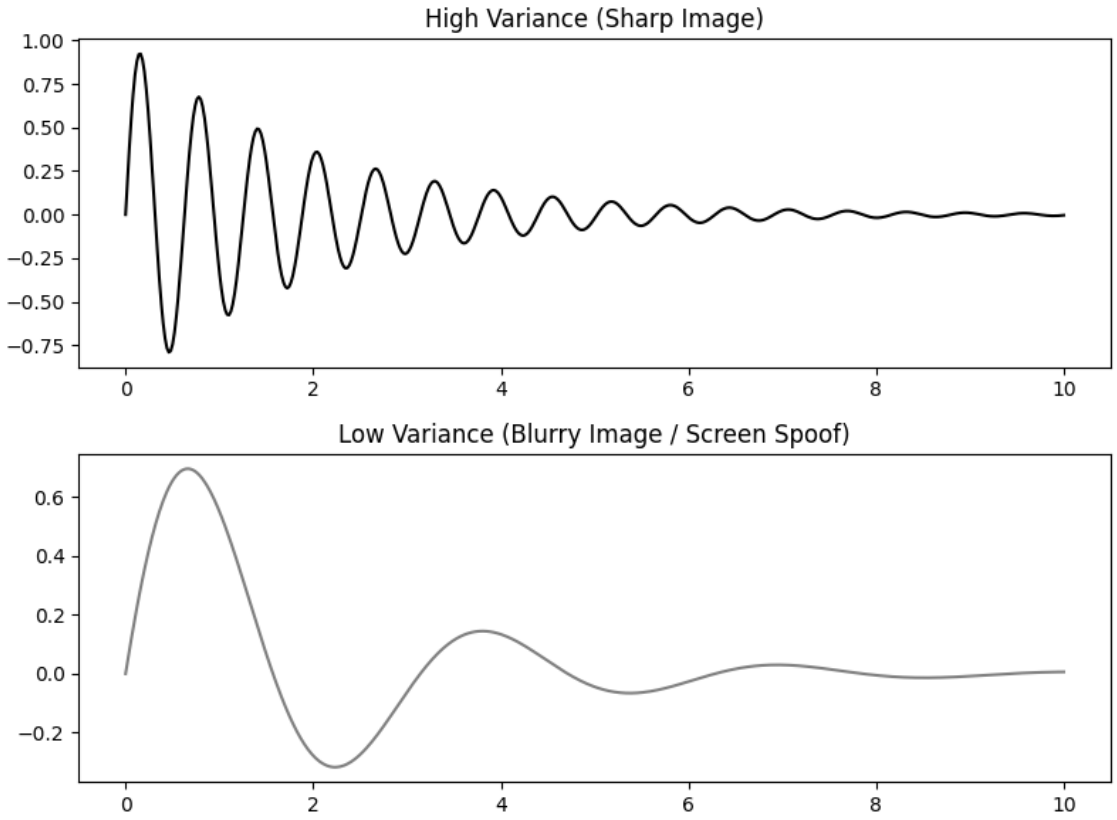
used to improve the performance of traditional biometric systems. Most soft biometrics can be easily collected and are actually collected during enrollment. Two main ethical issues are raised by soft biometrics. First, some of soft biometric traits are strongly cultural based; e.g., skin colors for determining ethnicity risk to support racist approaches, biometric sex recognition at the best recognizes gender from tertiary sexual characters, being unable to determine genetic and chromosomal sexes; soft biometrics for aging recognition are often deeply influenced by ageist stereotypes, etc. Second, soft biometrics have strong potential for categorizing and profiling people, so risking of supporting processes of stigmatization and exclusion. === Data protection of biometric data in international law === Many countries, including the United States, are planning to share biometric data with other nations. In testimony before the US House Appropriations Committee, Subcommittee on Homeland Security on "biometric identification" in 2009, Kathleen Kraninger and Robert A Mocny commented on international cooperation and collaboration with respect to biometric data, as follows: To ensure we can shut down terrorist networks before they ever get to the United States, we must also take the lead in driving international biometric standards. By developing compatible systems, we will be able to securely share terrorist information internationally to bolster our defenses. Just as we are improving the way we collaborate within the U.S. Government to identify and weed out terrorists and other dangerous people, we have the same obligation to work with our partners abroad to prevent terrorists from making any move undetected. Biometrics provide a new way to bring terrorists' true identities to light, stripping them of their greatest advantage—remaining unknown. According to an article written in 2009 by S. Magnuson in the National Defense Magazine entitled "Defense Department Under Pressure to Share Biometric Data" the United States has bilateral agreements with other nations aimed at sharing biometric data. To quote that article: Miller [a consultant to the Office of Homeland Defense and America's security affairs] said the United States has bilateral agreements to share biometric data with about 25 countries. Every time a foreign leader has visited Washington during the last few years, the State Department has made sure they sign such an agreement. === Likelihood of full governmental disclosure === Certain members of the civilian community are worried about how biometric data is used but full disclosure may not be forthcoming. In particular, the Unclassified Report of the United States' Defense Science Board Task Force on Defense Biometrics states that it is wise to protect, and sometimes even to disguise, the true and total extent of national capabilities in areas related directly to the conduct of security-related activities. This also potentially applies to Biometrics. It goes on to say that this is a classic feature of intelligence and military operations. In short, the goal is to preserve the security of 'sources and methods'. === Data security === The frequent use of biometric authentication for security and the permanence of an individuals biometrics make the security of biometric data crucial. ==== Events where biometric data was compromised ==== Office of Personnel Management data breach in 2015 Biostar 2 fingerprints leak in 2019 Taliban seizure of US biometric data in 2021 Afghan & Iraqi Fingerprints and Iris database ==== Legislation and governmental Action ==== Biometrics are considered personal information/data under multiple laws GDPR in the European Union became law in 2018 LGPD in Brazil became law in 2020 Protection of Personal Information Act in South Africa came into force in 2020 Personal Data Protection Act in Sri Lanka implementation started in 2023 ===== United States ===== The United States does not have a nationwide data privacy law that includes biometrics. Several states and local governments, led by the Illinois Biometric Information Privacy Act, have legislation regarding biometric data. The FTC has also taken actions to protect biometric data including against Facebook in 2019, charging they misrepresented their uses of facial

recognition technology. == Countries applying biometrics == Countries using biometrics include Australia, Brazil, Bulgaria, Canada, Cyprus, Greece, China, Gambia, Germany, India, Iraq, Ireland, Israel, Italy, Malaysia, Netherlands, New Zealand, Nigeria, Norway, Pakistan, Poland, South Africa, Saudi Arabia, Tanzania, Turkey, Ukraine, United Arab Emirates, United Kingdom, United States and Venezuela. Among low to middle income countries, roughly 1.2 billion people have already received identification through a biometric identification program. There are also numerous countries applying biometrics for voter registration and similar electoral purposes. According to the International IDEA's ICTs in Elections Database, some of the countries using (2017) Biometric Voter Registration (BVR) are Armenia, Angola, Bangladesh, Bhutan, Bolivia, Brazil, Burkina Faso, Cambodia, Cameroon, Chad, Colombia, Comoros, Congo (Democratic Republic of), Costa Rica, Ivory Coast, Dominican Republic, Fiji, Gambia, Ghana, Guatemala, India, Iraq, Kenya, Lesotho, Liberia, Malawi, Mali, Mauritania, Mexico, Morocco, Mozambique, Namibia, Nepal, Nicaragua, Nigeria, Panama, Peru, Philippines, Senegal, Sierra Leone, Solomon Islands, Somaliland, Swaziland, Tanzania, Uganda, Uruguay, Venezuela, Yemen, Zambia, and Zimbabwe. === India's national ID program === India's national ID program called Aadhaar is the largest biometric database in the world. It is a biometrics-based digital identity assigned for a person's lifetime, verifiable online instantly in the public domain, at any time, from anywhere, in a paperless way. It is designed to enable government agencies to deliver a retail public service, securely based on biometric data (fingerprint, iris scan and face photo), along with demographic data (name, age, gender, address, parent/spouse name, mobile phone number) of a person. The data is transmitted in encrypted form over the internet for authentication, aiming to free it from the limitations of physical presence of a person at a given place. About 550 million residents have been enrolled and assigned 480 million Aadhaar national identification numbers as of 7 November 2013. It aims to cover the entire population of 1.2 billion in a few years. However, it is being challenged by critics over privacy concerns and possible transformation of the state into a surveillance state, or into a Banana republic.§ The project was also met with mistrust regarding the safety of the social protection infrastructures. To tackle the fear amongst the people, India's supreme court put a new ruling into action that stated that privacy from then on was seen as a fundamental right. On 24 August 2017 this new law was established. === Malaysia's MyKad national ID program === The current identity card, known as MyKad, was introduced by the National Registration Department of Malaysia on 5 September 2001 with Malaysia becoming the first country in the world to use an identification card that incorporates both photo identification and fingerprint biometric data on a built-in computer chip embedded in a piece of plastic. Besides the main purpose of the card as a validation tool and proof of citizenship other than the birth certificate, MyKad also serves as a valid driver's license, an ATM card, an electronic purse, and a public key, among other applications, as part of the Malaysian Government Multipurpose Card (GMPC) initiative, if the bearer chooses to activate the functions. == See also == == Notes == == References == == Further reading == Biometrics Glossary – Glossary of Biometric Terms based on information derived from the National Science and Technology Council (NSTC) Subcommittee on Biometrics. Published by Fulcrum Biometrics, LLC, July 2013 Biometrics Institute - Explanatory Dictionary of Biometrics A glossary of biometrics terms, offering detailed definitions to supplement existing resources. Published May 2023. Delac, K., Grgic, M. (2004). A Survey of Biometric Recognition Methods. "Fingerprints Pay For School Lunch". (2001). Retrieved 2008-03-02. [1] "Germany to phase-in biometric passports from November 2005". (2005). E-Government News. Retrieved 2006-06-11. [2] Oezcan, V. (2003). "Germany Weighs Biometric Registration Options for Visa Applicants", Humboldt University Berlin.

Retrieved 2006-06-11. Ulrich Hottelet: Hidden champion – Biometrics between boom and big brother, German Times, January 2007. Dunstone, T. and Yager, N., 2008. Biometric system and data analysis. 1st ed. New York: Springer. == External links ==

4.2 Laplacian Variance (Blur Detection)

When an attacker presents a phone screen or a printed photograph, the image captured by the attendance camera is technically a 're-capture'. This process intrinsically loses high-frequency details. The system applies a Laplacian operator (second derivative of the image) to measure sharpness. If the variance of the Laplacian is low, the image is too blurry to be a physical person.



4.3 3D Geometry and the Nose Ratio

Since YuNet extracts 5 facial landmarks, the system can understand the 3D orientation of the head. When a user is asked to 'turn their head', a 2D photograph will merely shift laterally across the frame. A 3D human head, however, undergoes perspective distortion.

Ratio = (Nose_X - BoundingBox_Left) / BoundingBox_Width

When looking straight ahead, the ratio is ~0.5. As the user turns right, the nose moves closer to the right edge of the bounding box, shifting the ratio towards 0.8. If the system detects these dynamic

changes across video frames, liveness is verified.

Part V: Source Code Deconstruction

This section contains the verbatim source code of the EEE 451 project's AI utilities, accompanied by architectural analysis.

5.1 face_engine.py - Detection Integration

Source Code Path: C:\Users\USER\Desktop\EEE451 PROJECT\Backend\utils\face_engine.py

```
from __future__ import annotations

from threading import Lock
from typing import List, Optional, Tuple

import os
import numpy as np

import cv2

from config import Config
from utils.embeddings import compute_embedding, crop_face_xyxy

_engine = None
_lock = Lock()


def _check_sface_available() -&gt; bool:
    if not hasattr(cv2, "FaceDetectorYN") or not hasattr(cv2, "FaceRecognizerSF"):
        return False
    det_path = Config.OPENCV_DET_MODEL
    rec_path = Config.OPENCV_REC_MODEL
    if not os.path.isabs(det_path):
        det_path = os.path.join(os.path.dirname(os.path.dirname(__file__)), det_path)
    if not os.path.isabs(rec_path):
        rec_path = os.path.join(os.path.dirname(os.path.dirname(__file__)), rec_path)
    return os.path.exists(det_path) and os.path.exists(rec_path)


def _init_engine():
    det_path = Config.OPENCV_DET_MODEL
    rec_path = Config.OPENCV_REC_MODEL
    if not os.path.isabs(det_path):
        det_path = os.path.join(os.path.dirname(os.path.dirname(__file__)), det_path)
    if not os.path.isabs(rec_path):
        rec_path = os.path.join(os.path.dirname(os.path.dirname(__file__)), rec_path)

    detector = cv2.FaceDetectorYN.create(
        det_path,
        "",
        (Config.OPENCV_DET_SIZE, Config.OPENCV_DET_SIZE),
        Config.OPENCV_DET_THRESH,
        0.3,
        5000,
    )
    recognizer = cv2.FaceRecognizerSF.create(rec_path, "")
    return {"detector": detector, "recognizer": recognizer}


def get_engine():
    global _engine
    if not _check_sface_available():
```

```
        return None
    if _engine is not None:
        return _engine
    with _lock:
        if _engine is None:
            _engine = _init_engine()
    return _engine


def _face_area_xyxy(bbox) -&gt; float:
    x1, y1, x2, y2 = bbox
    return float(max(0.0, x2 - x1) * max(0.0, y2 - y1))

def _iou_xyxy(a, b) -&gt; float:
    ax1, ay1, ax2, ay2 = a
    bx1, by1, bx2, by2 = b
    ix1 = max(ax1, bx1)
    iy1 = max(ay1, by1)
    ix2 = min(ax2, bx2)
    iy2 = min(ay2, by2)
    iw = max(0.0, ix2 - ix1)
    ih = max(0.0, iy2 - iy1)
    inter = iw * ih
    if inter &lt;= 0.0:
        return 0.0
    a_area = max(0.0, ax2 - ax1) * max(0.0, ay2 - ay1)
    b_area = max(0.0, bx2 - bx1) * max(0.0, by2 - by1)
    union = max(1e-8, a_area + b_area - inter)
    return float(inter / union)


def _nms_faces(faces, iou_thresh: float):
    if faces is None:
        return faces
    if len(faces) == 0:
        return faces

    arr = np.asarray(faces)
    if arr.ndim == 1:
        arr = arr.reshape(1, -1)

    boxes = []
    scores = []
    for f in arr:
        x, y, fw, fh = f[:4]
        boxes.append((float(x), float(y), float(x + fw), float(y + fh)))
        scores.append(float(f[4]) if len(f) &gt; 4 else 1.0)

    idxs = np.argsort(scores)[::-1]
    keep = []
    while len(idxs) &gt; 0:
        i = int(idxs[0])
        keep.append(i)
        if len(idxs) == 1:
            break
        rest = idxs[1:]
        kept = []
        for j in rest:
            if _iou_xyxy(boxes[i], boxes[int(j)]) &lt;= iou_thresh:
                kept.append(int(j))
        idxs = np.asarray(kept, dtype=int)

    return arr[keep]


def extract_all_embeddings(image_bgr) -&gt; List[Tuple[np.ndarray, Tuple[int, int, int, int], fl
```

```
    """
    Returns a list of (embedding, bbox_xyxy, det_score).
    bbox is (x1, y1, x2, y2) int.
    """
    engine = get_engine()
    if engine is None:
        # Fallback: Haar detector + simple embeddings
        gray = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)
        detector = cv2.CascadeClassifier(
            cv2.data.haarcascades + "haarcascade_frontalface_default.xml"
        )
        faces = detector.detectMultiScale(
            gray, scaleFactor=1.2, minNeighbors=5, minSize=(70, 70)
        )
        out = []
        if faces is None or len(faces) == 0:
            return out
        for (x, y, w, h) in faces:
            xyxy = (int(x), int(y), int(x + w), int(y + h))
            face = crop_face_xyxy(image_bgr, xyxy)
            emb = compute_embedding(face)
            out.append((emb.astype(np.float32), xyxy, 1.0))
        return out

    detector = engine["detector"]
    recognizer = engine["recognizer"]

    h, w = image_bgr.shape[:2]
    detector.setInputSize((w, h))

    _, faces = detector.detect(image_bgr)
    out = []
    if faces is None:
        return out
    faces = _nms_faces(faces, Config.OPENCV_NMS_THRESH)

    for f in faces:
        x, y, fw, fh = f[:4]
        score = float(f[4]) if len(f) &gt; 4 else 1.0
        xyxy = (int(x), int(y), int(x + fw), int(y + fh))
        try:
            aligned = recognizer.alignCrop(image_bgr, f)
            emb = recognizer.feature(aligned)
            emb = emb.flatten().astype(np.float32)
            out.append((emb, xyxy, score))
        except Exception:
            continue
    return out


def extract_all_embeddings_with_landmarks(
    image_bgr,
) -&gt; List[Tuple[np.ndarray, Tuple[int, int, int, int], float, Optional[List[Tuple[float, floa
    """
    Returns a list of (embedding, bbox_xyxy, det_score, landmarks).
    landmarks: list of 5 (x,y) points or None if unavailable.
    """
    engine = get_engine()
    if engine is None:
        # Fallback: Haar detector + simple embeddings (no landmarks)
        gray = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)
        detector = cv2.CascadeClassifier(
            cv2.data.haarcascades + "haarcascade_frontalface_default.xml"
        )
        faces = detector.detectMultiScale(
            gray, scaleFactor=1.2, minNeighbors=5, minSize=(70, 70)
```

```
        )
        out = []
        if faces is None or len(faces) == 0:
            return out
        for (x, y, w, h) in faces:
            xyxy = (int(x), int(y), int(x + w), int(y + h))
            face = crop_face_xyxy(image_bgr, xyxy)
            emb = compute_embedding(face)
            out.append((emb.astype(np.float32), xyxy, 1.0, None))
        return out

    detector = engine["detector"]
    recognizer = engine["recognizer"]

    h, w = image_bgr.shape[:2]
    detector.setInputSize((w, h))

    _, faces = detector.detect(image_bgr)
    out = []
    if faces is None:
        return out
    faces = _nms_faces(faces, Config.OPENCV_NMS_THRESH)

    for f in faces:
        x, y, fw, fh = f[:4]
        score = float(f[4]) if len(f) &gt; 4 else 1.0
        xyxy = (int(x), int(y), int(x + fw), int(y + fh))

        landmarks = None
        if len(f) &gt;= 15:
            landmarks = [
                (float(f[5]), float(f[6])),   # left eye
                (float(f[7]), float(f[8])),   # right eye
                (float(f[9]), float(f[10])),  # nose
                (float(f[11]), float(f[12])), # left mouth
                (float(f[13]), float(f[14])), # right mouth
            ]

        try:
            aligned = recognizer.alignCrop(image_bgr, f)
            emb = recognizer.feature(aligned)
            emb = emb.flatten().astype(np.float32)
            out.append((emb, xyxy, score, landmarks))
        except Exception:
            continue
    return out


def extract_best_embedding(image_bgr) -&gt; Optional[Tuple[np.ndarray, Tuple[int, int, int, int]
    """
    Returns (embedding, bbox_xyxy, det_score) for the best face.
    Best = highest det_score, then largest area.
    """
    faces = extract_all_embeddings(image_bgr)
    if not faces:
        return None
    return max(faces, key=lambda it: (it[2], _face_area_xyxy(it[1])))


def extract_best_embedding_with_landmarks(
    image_bgr,
) -&gt; Optional[Tuple[np.ndarray, Tuple[int, int, int, int], float, Optional[List[Tuple[float,
    """
    Returns (embedding, bbox_xyxy, det_score, landmarks) for the best face.
    Best = highest det_score, then largest area.
    """
```

```
    faces = extract_all_embeddings_with_landmarks(image_bgr)
    if not faces:
        return None
    return max(faces, key=lambda it: (it[2], _face_area_xyxy(it[1])))


def embedding_from_detection(image_bgr, det) -&gt; Optional[np.ndarray]:
    """
    Compute an embedding from a YuNet detection row (with landmarks).
    Returns embedding vector or None.
    """
    engine = get_engine()
    if engine is None:
        return None
    recognizer = engine["recognizer"]
    try:
        aligned = recognizer.alignCrop(image_bgr, det)
        emb = recognizer.feature(aligned)
        return emb.flatten().astype(np.float32)
    except Exception:
        return None
```

Analysis: This file acts as the singleton wrapper for the OpenCV DNN module loading the ONNX models for YuNet and SFace. It handles coordinate translations, Non-Maximum Suppression (NMS) for overlapping boxes, and extraction of the 5 landmarks.

5.2 liveness.py - Anti-Spoofing Logic

Source Code Path: C:\Users\USER\Desktop\EEE451 PROJECT\Backend\utils\liveness.py

```
from __future__ import annotations

from typing import Dict, List, Optional, Tuple

import numpy as np

from config import Config
from utils.embeddings import blur_score, crop_face_xyxy


def _dist(a: Tuple[float, float], b: Tuple[float, float]) -&gt; float:
    ax, ay = a
    bx, by = b
    return float(((ax - bx) ** 2 + (ay - by) ** 2) ** 0.5)


def evaluate_liveness(
    image_bgr,
    bbox_xyxy: Tuple[int, int, int, int],
    landmarks: Optional[List[Tuple[float, float]]],
) -&gt; Dict[str, object]:
    """
    Heuristic liveness check using face size, eye distance, and blur.
    Returns:
    {
      "checked": bool,
      "score": float,
      "pass": bool,
      "details": { ... }
```

```
    }
    """
    if landmarks is None or len(landmarks) &lt; 2:
        return {"checked": False, "score": 0.0, "pass": False, "details": {"reason": "no_landmar

    x1, y1, x2, y2 = bbox_xyxy
    h_img, w_img = image_bgr.shape[:2]
    face_area = max(0, x2 - x1) * max(0, y2 - y1)
    img_area = max(1, w_img * h_img)
    face_ratio = face_area / img_area

    left_eye = landmarks[0]
    right_eye = landmarks[1]
    eye_dist = _dist(left_eye, right_eye)
    bbox_w = max(1, x2 - x1)
    eye_ratio = eye_dist / bbox_w

    face = crop_face_xyxy(image_bgr, bbox_xyxy)
    blur = blur_score(face)

    face_ratio_score = min(1.0, face_ratio / max(1e-6, Config.LIVENESS_MIN_FACE_RATIO))
    eye_ratio_score = min(1.0, eye_ratio / max(1e-6, Config.LIVENESS_MIN_EYE_DIST_RATIO))
    blur_score_norm = min(1.0, blur / max(1e-6, Config.BLUR_THRESHOLD))

    score = float((face_ratio_score + eye_ratio_score + blur_score_norm) / 3.0)
    passed = score &gt;= Config.LIVENESS_MIN_SCORE

    return {
        "checked": True,
        "score": score,
        "pass": passed,
        "details": {
            "face_ratio": face_ratio,
            "eye_ratio": eye_ratio,
            "blur_score": blur,
        },
    }


def _nose_ratio(
    bbox_xyxy: Tuple[int, int, int, int],
    landmarks: Optional[List[Tuple[float, float]]],
) -&gt; Optional[float]:
    if not landmarks or len(landmarks) &lt; 3:
        return None
    x1, y1, x2, y2 = bbox_xyxy
    nose_x, _ = landmarks[2]
    w = max(1.0, float(x2 - x1))
    return float((nose_x - x1) / w)


def evaluate_liveness_challenge(
    frames: List[Tuple[Tuple[int, int, int, int], Optional[List[Tuple[float, float]]]]],
    challenge: str,
) -&gt; Dict[str, object]:
    """
    Multi-frame challenge using landmark nose position:
    - challenge: "turn_left", "turn_right", "left_right"
    """
    if not frames:
        return {"ok": False, "pass": False, "reason": "no_frames"}

    ratios = []
    for bbox, landmarks in frames:
        ratio = _nose_ratio(bbox, landmarks)
        if ratio is None:
```

```
            continue
        ratios.append(ratio)

    if len(ratios) &lt; 2:
        return {"ok": False, "pass": False, "reason": "no_landmarks"}

    left = any(r &lt;= 0.5 - Config.LIVENESS_CHALLENGE_SHIFT for r in ratios)
    right = any(r &gt;= 0.5 + Config.LIVENESS_CHALLENGE_SHIFT for r in ratios)
    center = any(abs(r - 0.5) &lt;= Config.LIVENESS_CHALLENGE_SHIFT for r in ratios)

    if challenge == "turn_left":
        passed = left and center
    elif challenge == "turn_right":
        passed = right and center
    elif challenge == "left_right":
        passed = left and right and center
    else:
        return {"ok": False, "pass": False, "reason": "invalid_challenge"}

    return {
        "ok": True,
        "pass": bool(passed),
        "challenge": challenge,
        "ratios": ratios,
        "details": {"left": left, "right": right, "center": center},
    }
```

Analysis: This module implements the mathematical heuristics discussed in Part IV. The 'evaluate_liveness_challenge' function iterates over multiple video frames to detect dynamic shifts in the nose ratio, confirming a 3D subject.

5.3 embeddings.py - Storage & Metrics

Source Code Path: C:\Users\USER\Desktop\EEE451 PROJECT\Backend\utils\embeddings.py

```
import os
import time
from typing import Optional, Tuple

import cv2
import numpy as np


# ----------------------------
# Paths
# ----------------------------
BASE_DIR = os.path.dirname(os.path.dirname(os.path.abspath(__file__)))
FACES_DIR = os.path.join(BASE_DIR, "dataset", "faces")
EMB_DIR = os.path.join(BASE_DIR, "embeddings", "students")


def ensure_dir(path: str) -&gt; None:
    os.makedirs(path, exist_ok=True)


# ----------------------------
# Face detection (Haar)
# ----------------------------
def crop_face(image_bgr, box, pad: int = 12):
```

```
    x, y, w, h = box
    h_img, w_img = image_bgr.shape[:2]

    x1 = max(0, x - pad)
    y1 = max(0, y - pad)
    x2 = min(w_img, x + w + pad)
    y2 = min(h_img, y + h + pad)

    return image_bgr[y1:y2, x1:x2]


def crop_face_xyxy(image_bgr, box_xyxy, pad: int = 12):
    x1, y1, x2, y2 = box_xyxy
    h_img, w_img = image_bgr.shape[:2]

    x1 = max(0, int(x1) - pad)
    y1 = max(0, int(y1) - pad)
    x2 = min(w_img, int(x2) + pad)
    y2 = min(h_img, int(y2) + pad)

    return image_bgr[y1:y2, x1:x2]


def blur_score(image_bgr) -&gt; float:
    gray = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2GRAY)
    return float(cv2.Laplacian(gray, cv2.CV_64F).var())


# ------------------------------
# Embedding (simple, works now)
# ------------------------------
def compute_embedding(face_bgr, size: int = 64) -&gt; np.ndarray:
    """
    Deprecated fallback embedding (used only if InsightFace is unavailable).
    """
    gray = cv2.cvtColor(face_bgr, cv2.COLOR_BGR2GRAY)
    resized = cv2.resize(gray, (size, size), interpolation=cv2.INTER_AREA)
    vec = resized.astype(np.float32).reshape(-1)
    norm = np.linalg.norm(vec) + 1e-8
    return vec / norm


def student_face_dir(student_id: str) -&gt; str:
    path = os.path.join(FACES_DIR, student_id)
    ensure_dir(path)
    return path


def student_emb_dir(student_id: str) -&gt; str:
    path = os.path.join(EMB_DIR, student_id)
    ensure_dir(path)
    return path


def save_face_and_embedding(
    student_id: str,
    view_type: str,
    face_bgr,
    embedding: np.ndarray,
    model_name: str = "insightface",
    max_samples: int = 8,
):
    """
    Saves:
    - cropped face image: dataset/faces/&lt;student_id&gt;/&lt;view_type&gt;_&lt;ts&gt;.jpg
    - embeddings file: embeddings/students/&lt;student_id&gt;/&lt;view_type&gt;.npy
```

```
    IMPORTANT (Batch 11):
    - &lt;view_type&gt;.npy now stores MANY embeddings:
      shape = (K, D)
      so we can recognize better in different lighting/angles.
    - We keep ONLY the latest `max_samples`.
    """

    ts = int(time.time() * 1000)

    face_path = os.path.join(student_face_dir(student_id), f"{view_type}_{ts}.jpg")
    cv2.imwrite(face_path, face_bgr)

    emb = embedding.astype(np.float32)  # shape (D,)
    emb_path = os.path.join(student_emb_dir(student_id), f"{view_type}.npy")

    if os.path.exists(emb_path):
        try:
            old = np.load(emb_path)
            # old can be (D,) or (K,D)
            if old.ndim == 1:
                old = old.reshape(1, -1)
            new = np.vstack([old, emb.reshape(1, -1)])
            # keep last max_samples
            if new.shape[0] &gt; max_samples:
                new = new[-max_samples:, :]
            np.save(emb_path, new)
        except Exception:
            # if file corrupt, overwrite
            np.save(emb_path, emb.reshape(1, -1))
    else:
        np.save(emb_path, emb.reshape(1, -1))

    return face_path, emb_path, model_name, emb


def delete_student_face_data(student_id: str) -&gt; dict:
    """
    Remove all stored face images and embeddings for a student.
    Returns counts for files removed (best effort).
    """
    import shutil

    emb_dir = os.path.join(EMB_DIR, student_id)
    face_dir = os.path.join(FACES_DIR, student_id)

    emb_files = 0
    face_files = 0

    if os.path.exists(emb_dir):
        for _, _, files in os.walk(emb_dir):
            emb_files += len(files)
        shutil.rmtree(emb_dir, ignore_errors=True)

    if os.path.exists(face_dir):
        for _, _, files in os.walk(face_dir):
            face_files += len(files)
        shutil.rmtree(face_dir, ignore_errors=True)

    return {
        "embeddings_deleted": int(emb_files),
        "faces_deleted": int(face_files),
    }
```

Analysis: This file manages the extraction and persistence of the 128-dimensional vectors. It saves them as Numpy arrays (.npy files) allowing for instantaneous loading and dot-product calculations

during attendance verification.

END OF DOCUMENT