Literature and Technology Review

Saahil Shihaz

December 3, 2021

Contents

1 Mobile Machine Learning

The potential of smartphones have still not been fully realised. With advancements in machine learning we can take the capabilities of smartphones to the next level by leveraging the advanced hardware within them to carry out complex tasks. This section will identify the key milestones within smartphone technology and how it leads us to integrating machine learning to make full use of the hardware.

Figure 1: Samsung Galaxy S21 Ultra 5G: Boasting an impressive array of camera sensors on the back (Three, 2021)

.

1.1 Evolution

Firstly, we want to look at the last 10 years of technological advancements within the smartphone space. With this information, we can hopefully gain some insight into the how much their capabilities have evolved. To do this, we will collect key aspects of specification data from the Samsung Galaxy flagship line of smartphones. Samsung currently hold the top spot in global market share at 20.8% as of Q3' 2021, this position is typically held by Apple or Samsung and can vary from a quarter to quarter basis (O'Dea, 2021).

Phone (Year)	Processor	Storage (GB)	Memory (GB)	Cameras (MP)
S2 (2011)	Dual-core 1.2 GHz	32	1	8
S3 (2012)	Quad-core 1.4 GHz	64	1	8
S4 (2013)	Octa-Core (4x1.6 GHz, 4x1.2 GHz)	64	2	13
S5 (2014)	Quad-Core 2.5 GHz	32	2	16
S6 edge+ (2015)	Octa-Core (4x2.1 GHz, 4x 1.5 GHz)	64	4	16
S7 edge (2016)	Octa-Core (4x 2.3 GHz, 4x 1.6 GHz)	128	4	12
S8+ (2017)	Octa-Core (4x 2.35 GHz, 4x 1.9 GHz)	128	6	12
S9+ (2018)	Octa-Core (4x 2.8 GHz, 4x 1.7 GHz)	256	6	12/12
S10+ (2019)	Octa-Core (4x 2.84 GHz, 4x 1.78 GHz)	1024	12	12/12/16
S20 Ultra 5G (2020)	Octa-Core (1x 2.84 GHz, 3x 2.42 GHz, 4x 1.8 GHz)	512	16	0.3/12/48/108
S21 Ultra 5G (2021)	Octa-Core (1x 2.84 GHz, 3x 2.42 GHz, 4x 1.8 GHz)	512	16	10/10/12/108

Table 1: All data is sourced from GSMArena, 2021

We can see a clear increase in smartphone capability in multiple categories like the processor speed, core count, storage, memory and camera capabilities. There are of course may more different areas that are not listed like sensors, screen size and battery life which have also seen massive improvements over the last 10 years. We have only been seeing improvements in this space, therefore we can assume that we will continue to see improvements in the near future. What we must also consider is price and accessibility, flagship smartphones represent the top of the line offerings from each smartphone manufacturer and are of course priced as such. Low-end to mid-end smartphones are still more capable than their predecessors albeit their spec sheet may not be as impressive as high-end versions in the same generation. Therefore, we must ensure some level of scalability within our machine and deep learning processes. How can we make sure our processes can run efficiently on lower end hardware as well as high end hardware? What we can analyse is; how is the hardware in smartphones being utilised currently, as a scientific tool?

Kulendran et al. (2014), highlights how improvements in smartphones have created a boom in the number of smartphone based application designed to aid surgeons and patients in multiple facets of the medical industry like plastic, orthopaedic and general surgery. They conduct an expansive review of different solutions and analyse how the evolution of smartphones got them to the point that make them extremely useful as a tool to aid us. Ultimately, what this project aims to do is provide a robust software solution to recognise flower species on a smartphone, but we cannot ignore the fact that smartphones have come a long way in the hardware and operating system space to allow us to even conceive of a system.

1.2 Smartphones as a scientific tool

Wright (2017), goes into extensive detail into recent research that helped doctors use smartphones as a tool to analyse cells, DNA in tissue and blood samples. This aids in specifically, the process of diagnosing. They analyse use cases where researchers make use of the smartphone camera as well as some additional apparatuses to analyse things like cells. An example where a smartphone with a 3D printed microscope was used to highlight cost efficiency and the portability of the apparatus. The smartphones camera assisted by the microscope was used to take multiple photos of the tissue sample which could then be processed by an algorithm and analysed, all within the smartphone. It was highlighted that a typical scenario would include using a $10,000 to $50,000 high end microscope within a lab, therefore having this type of capability in a remote environment at the fraction of the cost really highlights the benefit of smartphones.

A more relevant example comes in the form of an application called Bilicam that consisted of a solution to identify jaundice within new-borns. Jaundice is a medical condition that can be identified by the yellow discoloration of skin (Greef et al., 2014). The system consisted of just the smartphone with the

application and a card of colours used for calibration. The research specifies that the chemical by-product of Bilirubin causes Jaundice and that there are scenarios where new-borns can have too much of it, this can cause irreversible brain damage and even death in some cases. They proposed a smartphone system that uses a picture of a new born which is then uploaded to a cloud service to be processed by a machine learning algorithm. It should then output a predicted Bilirubin level. Their rationale for this system was to have a solution where you could easily screen for new-borns that could potentially have dangerous levels of jaundice, without conducting blood tests. This could mean when the new-born has already left the hospital and therefore getting an accurate lab based diagnosis is no longer possible. The analysis system used the card to essentially ensure that the colour data was accurate as environmental factors such as lighting can affect the outcome. It then passes on the image data into a cloud based machine learning process that consisted of different regression algorithms, each algorithm's output is then combined to produce a final output for Bilirubin levels. Their research showed that the final product was effective at estimating the Bilirubin levels by comparing it to the Transcutaneous bilirubin (TcB) test. This is test used to measure Bilirubin without gathering blood from the patient (Centre, n.a.). The paper expanded on the fact that it still required to test it's findings on smartphones from different brands and that it required an internet connection in order to function. Future work would need to investigate the feasibility of having an on-device solution, this is something that this project will hope to investigate. Interestingly, there is no mention of deep learning within this experiment and they chose to go with traditional machine learning algorithms like k-Nearest neighbour. This could simply be because of the year (2014) that this experiment was conducted and therefore Deep Learning was not considered. This is backed up by the Google trends graph which maps out interest in a search term over time which sees a rise in interest after 2014 (Google, 2021a).

1.3 Where does this lead us to, today?

ML and AI has become such an important part of smartphones that manufacturers now have dedicated processors for ML and AI tasks. Google includes a Tenor Processing Unit (TPU) in their Pixel line of phones (Triggs and Simons, 2021). Samsung, Qualcomm and Apple use their own solutions for machine learning processing by having their own bespoke processors. These processors are used to compute specific actions that require the decision making and accuracy capabilities of machine learning. Google Tensor in particular aids tasks such as speech recognition that is accurate but not taxing, therefore saving battery life. Tensor also applies to processing photographs and provides additional features to videos (Gupta, 2021). With such a focus on smartphones, to the point that they get dedicated hardware for ML, we should be seeing a huge increase in applications that integrate ML in some way, as well as the entire process of designing and implementing such solutions being carried out more rapidly, as developers learn to leverage the hardware.

2 Computer Vision

Since we are working with analysing images, the area of computer vision plays a big part in our research. In order to identify flower species we must first discuss techniques to analyse the incoming image data to make predictions using ML and DL.

2.1 History

Szeliski (2011) outlines significant occurrences in each decade starting from the 70s, thought to be the beginning of computer vision, all the way through to the 2000s. In the early 70s, researchers sought to emulate human intelligence in a machine by first solving the visual problem. It was hypothesised that if a computer could first recognize objects in the real world that it could then move onto the next step of using reasoning and problem solving at a high level. The first processes conducted to understand the 3D world were to extract edges to recognize 3D objects from 2D lines in an image.

The 80s were described to have a lot more focus on mathematical techniques for analysing scenes. Various algorithms and models were conceived as well as improvements in the contour and edge detection space. Researchers found that a lot of these algorithms could be thought of as "optimization problems" when they were described using the same mathematical framework.

We see more improvements in the field during the 90s including the production of 3D surfaces, tracking and image segmentation. However, what is probably more relevant to this project is statistical learning techniques that also started to appear during this decade. In 1991, we see a paper by Turk and Pentland (1991) that described the concept of "eigenfaces". These are the product of converting images of faces into feature images. These feature images are essentially the training set. Recognition occurs "by projecting a new image into the sub-space spanned by the eigenfaces". The new face is then classified by comparing it's position relative to the known set of faces. Emphasis was placed on the limiting the scope of the allowed images, as such the system was trained and ready to accept profile straight-on images of the subject. In addition to that, they aimed to have the system compute a result in a reasonable time, which of course, is one of the goals of this project. The research hoped to improve on it's predecessors that used, at the time, traditional methods of recognising features such as eyes', noses and mouths and their relative position to each other. The work done with eigenfaces shows great similarities with the machine learning techniques we see today, by essentially creating feature vectors and comparing the distance of known vectors in the same space.

Szeliski (2011) continues with their insight into the 2000s where see the various improvements like more efficient algorithms and what finally dominates the latter half of the 2000s; applying machine learning techniques to computer

vision to aid visual recognition research.

3 Machine Learning

This project will be using ML techniques to compare efficiency and accuracy to the more evolved deep learning. What we must first consider is how machine learning works in the context of computer vision. Camastra and Vinciarelli (2015) summarise this and broke down ML development around three primary research points:

- Task-Oriented Studies, improving performance of learning systems in a predetermined set of tasks.
- Cognitive Simulation, emulating the human brain and designing processes around the human thought process.
- Theoretical Analysis, "the theoretical investigation of possible learning methods and algorithms independently of application domain".

They also produce a taxonomy to represent the balance of two entities they describe: the "teacher" and the "learner". The teacher, being the programmer, the one that designs the learning process and the learner being the computer system. The idea of inference is also introduced where a system can derive knowledge from previous observations. The taxonomy breaks down the amount of work that both the "learner" and the "teacher" need to do into four categories: Rote Learning, Learning from instruction, Learning by analogy and Learning from examples.

What we are more interested in is learning from examples where the "learner" infers the most out of the other categories in the taxonomy. The idea of the "learning problem" is introduced where the system needs to find a "general rule that explains the data given only a sample of limited size". Learning techniques are broken down into four more categories: Supervised learning, Reinforcement learning, Unsupervised learning, Semi-supervised learning.

Zhu (2005) highlights semi-supervised learning in their survey as the combination of supervised and unsupervised learning where we use both labelled and unlabelled data for training of the classifier. They point to the survey done by Seeger (2000) in particular that provides more insight into the concept of semi-supervised learning. Their rationale for the concept in general was applying the ability for a system to make predictions based on knowledge it doesn't have. A supervised system has all labelled data to aid it's training therefore it's basis on making predictions is described as a "security belt" by Seeger. The model will basically make predictions within its limited scope, what we would call "overfitting" (Dietterich, 1995). Unsupervised learning heavily relies on prior assumptions for their final result this is because it doesn't have a knowledge base to rely on. By using a balanced combination of both implementations we

can "balance the impact of prior assumptions". Seeger also highlights the fact that labelling the data is a taxing process, fortunately for us, existing data sets already exist with labelled and unlabelled images for flower species which will be useful for training. Therefore, a semi-supervised approach is feasible for the ML approach of this project.

3.1 Classifiers

Fortunately, there is no shortage of types of classifiers in the ML space. Mohammed (2017) covers the most popular ones in good detail such as Naïve Bayes, k-Nearest Neighbour and Support Vector Machines (SVM). Starting with Naïve Bayes, this is a supervised classifier based on probability that assumes all attributes are independent:

$$P(c|E) = \frac{P(E|c)P(C)}{P(E)} \tag{1}$$

Where E is classified as the class $C = +$ if and only if

$$BC(E) = \frac{P(C = +|E)}{P(C = -|E)} \geq 1$$

BC is our Bayesian classifier, + and – are two separate classes (Zhang, 2004).

Zhang (2004) states that Naïve Bayes is surprisingly superb in classification and demonstrates the classifier based version of it in Equation 1. They explore the optimal conditions of Naïve Bayes and propose that it is most optimal when the dependencies among attributes cancel out since Naïve Bayes works best when each attribute is independent, this is relevant as we want to explore optimisation in the project to ensure the highest level of efficiency when making the flower predictions.

Mohammed (2017) states that K-Nearest Neighbours (KNN) is one of the "simplest" of all the ML algorithms. Rosebrook (2016) discusses how to implement an image classifier using KNN where we can convert an image into a set of feature vectors on a graph, any new points get classified based on the k number of nearest neighbouring points, there's no real learning in this process, just the calculation of where the nearest points are, based on (usually Euclidean) distance.

Noble (2006) describes SVM as a way to tackle binary classifications, which means in the context of flower classification it only really answers questions like "is it Flower A?". They state that you would need to train multiple "one-versus-all" classifiers. For this project, that may not be appropriate if we want to support a large number of flower species.

ML techniques are certainly not useless and can still provide results, however,

the research in the space has evolved to a new level, aiming to improve upon these traditional ML techniques in all evaluation categories. This is where Deep Learning (DL) comes in.

4 Deep Learning

DL will of course be our alternative approach to recognizing flower species. Where ML is basically our baseline, our DL implementation should hopefully highlight how much better it is compared to the ML approach.

4.1 Neurons and Perceptrons

Scarpino (2018) introduces the concept of Perceptrons in their book about using TensorFlow to implement DL. First, we must discuss neurons and how they relate to understanding the foundation of DL.

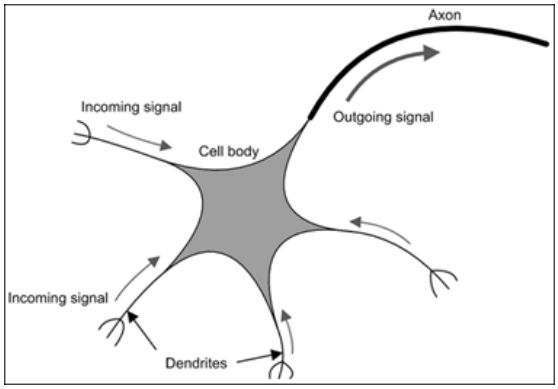

Figure 2: Simple diagram of a neuron (Scarpino, 2018)

.

What they choose to highlight in particular are three points that describe a neurons functionality (and ultimately how it relates to perceptrons):

- A neuron receives one or more incoming signals and produces one outgoing signal.
- A neuron's output can serve as the input of another neuron.
- Every neuron has a threshold, and the neuron won't produce output until its electricity exceeds the threshold.

This page by Anon (n.d.) highlights a brief history of perceptrons, though it serves a starting point to learn more about the concept. Perceptrons were coined by Frank Rosenblatt in 1962 (Rosenblatt, 1961), though his research is a bit outdated for our analysis, therefore here is a more modern version:

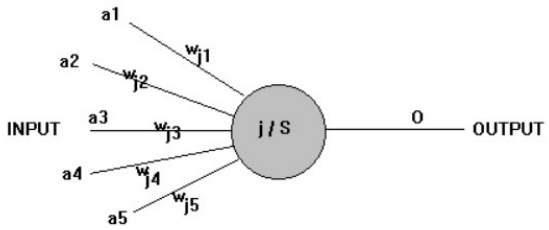

Figure 3: Diagram of a perceptron (Anon, n.d.(b))

.

Each input on the left is weighted and the summed within the circle node. If the summation meets a certain threshold, the output will be 1, if it doesn't meet the threshold then a 0 is outputted (Scarpino, 2018). Scarpino highlights some improvements to the model that was made including the weights that we discussed earlier as well as additional biases assigned with the incoming signals and an "activation function" that generates the output signal. Scarpino goes further by linking activation functions directly with in built TensorFlow functions that carry out the same task. Making it quite useful to understand the link between the TensorFlow API and the underlying DL context. Once we start linking perceptrons and arranging them into layers we get a neural network as shown here:

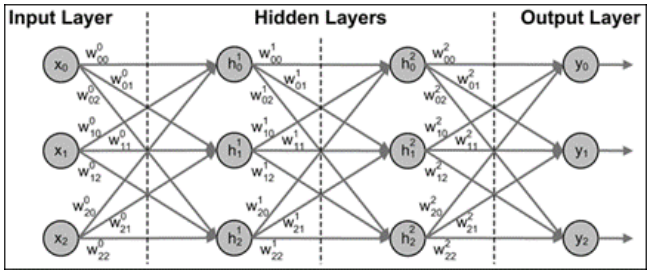

Figure 4: Diagram of a layered network of perceptrons (Scarpino, 2018)

.

4.2 Mobile Deep Learning

Goodfellow, Bengio, and Courville (2015) go into detail about Deep Learning from the concept of perceptrons to modern implementations. Something they talk about that is interesting is the increasing data set and model sizes over time, which is quite applicable to the project since we are working with modern mobile hardware. They discuss how the increasing capabilities of computer hardware have led to the development of larger models and that neural networks tend to

double in size roughly every 2.4 years. They predict that the trend will continue further on in the future. What is also relevant is the dataset sizes. Storing datasets take up storage space, they state that a deep learning algorithm (as of 2015) is stated to perform at acceptable levels with "around 5,000 labelled examples per category, and will match or exceed human performance when trained with a dataset containing at least 10,000,000 labelled examples. That is of course, extremely large and is most definitely going to take up a lot of storage. Therefore, the book does re-iterate the earlier point of making use of unlabelled data like with semi-supervised learning. Goodfellow, Bengio, and Courville (2015) dig deep into the subject of DL and explain subjects from the applied maths to the modern practices of DL and the research in the field. This can prove useful in fully understanding the various processes in place including optimisations and ways to increase accuracy that we will need to consider when designing a DL model for the mobile application.

4.3 Convolutional Neural Networks

Bengio, Goodfellow and Courville (2015) and Scarpino (2018) go into detail about CNNs, Scarpino in particular is a useful source on how it works with image classification in TensorFlow. However, it's useful to have some sort of starting point for the subject. Saha (2018) highlights the key features of a CNN and their purposes such as the individual layers: convolutional (kernel), pooling and classification (see Figure 5).

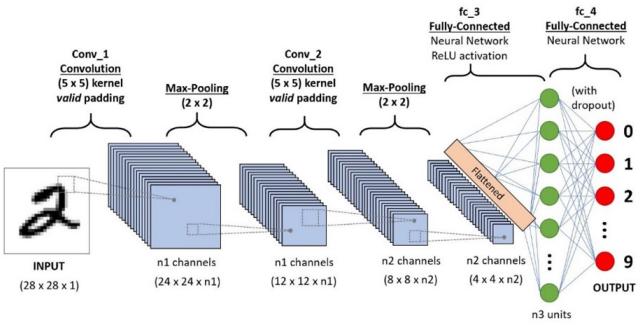

Figure 5: Example of a CNN process (Saha, 2018)

.

They also highlight a feature of CNNs that make images easier to process, where it reduces the size of images "into a form that is easier to process, without losing features which are critical for getting a good prediction". This helps with our scalability approach when it comes to dataset sizes in particular. The ELI5 (Explain Like I'm 5) format is quite useful and allows us to highlight the key points of each layer (Saha, 2018):

- Convolution: Applying the kernel to "extract the high-level features".
- Pooling: Reduces the spatial size of the output from the convolution. Decreases the "computational power" needed for data processing. Additionally, highlights features that are dominant.
- Classification: The pooling output is converted into column vectors and fed into a feed-forward neural network where the model is able to distinguish between features and classify them.

Saha (2018) also highlights that there are actually different implementations of CCNs, therefore they may not function in exactly the same format. Dive into Deep Learning (Anon, n.d.(a)) has a run down of multiple CNN types starting with LeNet-5 and more modern approaches like AlexNet, VGG, NiN, GoogLeNet, etc. We can also see how to implement them using TensorFlow which will prove useful when implementing our own solution for flowers.

4.4 ML vs DL

DL is an obvious evolution from ML, but it is worth highlighting the key differences for clarity because ultimately this project will compare how ML and DL compete with each other. Kavlakoglu (2020) breaks down how DL is different from ML. They highlight that DL takes the initiative by automating feature extraction to lessen human intervention and that ML is more reliant on humans, where humans normally define the characteristics to look out for, as well as their priorities. DL doesn't require a labelled dataset and can determine

features without it. DL is stated to "require more data points to improve its accuracy" compared to the ML counterpart.

Xin et al. (2018) goes into depth about the key differences when discussing approaches to ML and DL in the context of cybersecurity. However, the same reasoning can be applied in our situation. The key points they highlight are:

- Data dependencies: DL performs better with larger datasets like mentioned earlier as well as ML outperforming DL with smaller data sets.
- Hardware dependencies: DL requires a lot of matrix calculations and therefore a Graphical Processing Unit (GPU) can be used to optimise these processes. Note that mobile hardware do contain GPU hardware but they are not on the same scale as dedicated GPUs you find in PC hardware. Therefore, it will be interesting to see how DL fares against ML when we keep this hardware dependency in mind.
- Feature processing: Once again iterating on the point mentioned before, DL can extract features directly from the data and requires less human intervention.
- Execution time: DL algorithms take a lot longer to train compared to ML, this is dependant on the amount of data.
- Interpretability: Because of the complexity of DL it is hard to determine how a DL algorithm generated a result, whereas ML is more clearer.

I have summarised the key points, but they go into much more detail which could be helpful in the evaluation stage of comparing the two approaches of ML and DL.

5 Evaluation

One of the key points of the project is having to evaluate the ML and DL approach, what we haven't discussed yet however, is how do we go about doing this?

Williams, Zander, and Armitage (2006) identify a process named "k-fold cross validation" where the data set is "divided into k subsets". One of the subsets is used to test the classifier and the rest (k-1) subsets is used as the training set. They use three performance metrics to test their ML systems: accuracy, precision and recall. Accuracy being the percentage of correct decisions over the total number of test instances. Precision and recall are a bit more complex. Fortunately, Shung (2018) demonstrates how these two differ to accuracy. Precision is the number of instances that are correctly determined over the total number of instances that are guessed, this is made up of correctly guessed instances as well as instances that are incorrectly guessed. Recall is the number of

correctly predicted instances over the true number of instances in the class. In addition to these evaluation methods, Williams, Zander, and Armitage (2006) outline measuring CPU and memory usage. Fortunately, TensorFlow (lite) contains benchmarking tools for us to measure: Initialization time, inference time of warmup state/steady state, memory usage during initialization time and overall memory usage (Google, 2021b). The TensorFlow (lite) guide also contains tutorials on how to choose the best model for the task by comparing model size and accuracy of different models as well as the time it takes to make the prediction. The lite version is designed specifically for mobile and internet of things (IoT) hardware, so the additional tools will prove useful for the project later when we are at the evaluation stage.

Chockwanich and Visoottiviseth (2019) use the same evaluation methods outlined when comparing different DL models implemented in TensorFlow. They also look at CPU usage percentages and processing time. They were able to make a clear conclusion of which model is better by evaluating all factors. A term called f1-score was also mentioned in their analysis. Shung (2018) also explains the relevancy of f1 score, it is essentially a way to determine a "balance between precision and recall". Korstanje (2021) discusses the F1 score and it's purpose in providing a better accuracy statistic that accounts for "imabalanced data", this is when you don't have a good balance of data for each class and therefore the classifier makes inacurate predictions heavily skewed towards classes you have more data for. The F1 score is calculated by:

$$\frac{2 * Precision * Recall}{Precision + Recall}$$

6 Final Thoughts

The review has really highlighted the progression ML and DL from the early concepts and how everything eventually fit together to form what we know today. The area will keep getting more exciting as we learn to optimise our current algorithms, come up with new ones and make use of advancing hardware capabilities. ML and DL is getting more and more accessible as manufacturers allow the use of their specialised hardware to developers who can make use of APIs built specifically for these tasks. Overall, we have built on the solid foundations of previous research and it's interesting to see how the field develops in the future. The project hopes to aid the field by investigating the ML and DL approaches in the mobile format as well as explore why we see certain results from the evaluation.

References

Anon, n.d.(a). *Convolutional neural networks* [Online]. Available from: `https://d2l.ai/chapter_convolutional-neural-networks/lenet.html` [Accessed December 2, 2021].

Anon, n.d.(b). *History of the perceptron* [Online]. Available from: https://home.csulb.edu/~cwallis/artificialn/History.htm [Accessed November 30, 2021].

Camastra, F. and Vinciarelli, A., n.d. *Machine learning for audio, image and video analysis: theory and applications*. 2nd ed. 2015, Advanced Information and Knowledge Processing. London: Springer London.

Centre, T.M., n.a. *Transcutaneous bilirubin measurement* [Online]. Available from: https://hhma.org/healthadvisor/pa-tranbili-pep/ [Accessed November 30, 2021].

Chockwanich, N. and Visoottiviseth, V., 2019. Intrusion detection by deep learning with tensorflow. *2019 21st international conference on advanced communication technology (icact)* [Online], pp.654–659. Available from: https://doi.org/10.23919/ICACT.2019.8701969.

Dietterich, T., 1995. Overfitting and undercomputing in machine learning. *Acm comput. surv.* [Online], 27(3), pp.326–327. Available from: https://doi.org/10.1145/212094.212114.

Goodfellow, I., Bengio, Y., and Courville, A., 2016. *Deep learning*. MIT press.

Google, 2021a. *Google trends* [Online]. Available from: https://trends.google.com/trends/explore?cat=1227&date=all&q=Deep%20Learning [Accessed November 30, 2021].

Google, 2021b. *Performance measurement* [Online]. Available from: https://www.tensorflow.org/lite/performance/measurement [Accessed December 2, 2021].

Greef, L. de, Goel, M., Seo, M., Larson, E., Stout, J., Taylor, J., and Patel, S., 2014. Bilicam: using mobile phones to monitor newborn jaundice. eng. *Proceedings of the 2014 acm international joint conference on pervasive and ubiquitous computing*, UbiComp '14. ACM, pp.331–342.

GSMArena, 2021. *Gsm arena* [Online]. Available from: https://www.gsmarena.com/ [Accessed November 30, 2021].

Gupta, M., 2021. *Google tensor is a milestone for machine learning* [Online]. Available from: https://blog.google/products/pixel/introducing-google-tensor/ [Accessed November 30, 2021].

Kavlakoglu, E., 2020. *Ai vs. machine learning vs. deep learning vs. neural networks: what's the difference?* [Online]. Available from: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks [Accessed December 2, 2021].

Korstanje, J., 2021. *The f1 score* [Online]. Available from: https://towardsdatascience.com/the-f1-score-bec2bbc38aa6 [Accessed December 2, 2021].

Kulendran, M., Lim, M., Laws, G., Chow, A., Nehme, J., Darzi, A., and Purkayastha, S., 2014. Surgical smartphone applications across different platforms: their evolution, uses, and users. *Surgical innovation*, 21(4), pp.427–440.

Mohammed, M., 2017. *Machine learning : algorithms and applications*. eng. 1st edition. Boca Raton: CRC Press.

Noble, W.S., 2006. What is a support vector machine? eng. *Nature biotechnology*, 24(12), pp.1565–1567.

O'Dea, S., 2021. *Global smartphone market share from 4th quarter 2009 to 3rd quarter 2021* [Online]. Available from: https://www.statista.com/statistics/271496/global-market-share-held-by-smartphone-vendors-since-4th-quarter-2009/ [Accessed November 30, 2021].

Rosebrook, A., 2016. *K-nn classifier for image classification* [Online]. Available from: https://www.pyimagesearch.com/2016/08/08/k-nn-classifier-for-image-classification/ [Accessed December 2, 2021].

Rosenblatt, F., 1961. *Principles of neurodynamics. perceptrons and the theory of brain mechanisms*. (Technical report). Cornell Aeronautical Lab Inc Buffalo NY.

Saha, S., 2018. *A comprehensive guide to convolutional neural networks — the eli5 way* [Online]. Available from: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 [Accessed December 2, 2021].

Scarpino, M., 2018. *Tensorflow for dummies*. eng, For dummies. Newark: Wiley.

Seeger, M., 2000. Learning with labeled and unlabeled data.

Shung, K.P., 2018. *Accuracy, precision, recall or f1?* [Online]. Available from: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9 [Accessed December 2, 2021].

Szeliski, R., 2011. *Computer vision : algorithms and applications*. eng. 1st ed. 2011., Texts in Computer Science. London: Springer London : Imprint: Springer.

Three, 2021. *Samsung galaxy s21 ultra 5g* [Online]. Available from: http://www.three.co.uk/samsung/galaxy-s21-ultra-5g?colour=phantom%20black&memory=128&paym=true [Accessed December 2, 2021].

Triggs, R. and Simons, H., 2021. *Google tensor vs snapdragon 888 series: how the pixel 6's chip shapes up* [Online]. Available from: https://www.androidauthority.com/google-tensor-vs-snapdragon-888-3025332/ [Accessed November 30, 2021].

Turk, M.A. and Pentland, A.P., 1991. Face recognition using eigenfaces. IEEE Computer Society, pp.586–587.

Williams, N., Zander, S., and Armitage, G., 2006. A preliminary performance comparison of five machine learning algorithms for practical ip traffic flow classification. *Sigcomm comput. commun. rev.* [Online], 36(5), pp.5–16. Available from: `https://doi.org/10.1145/1163593.1163596`.

Wright, A., 2017. Smartphone science. *Commun. acm* [Online], 61(1), pp.18–20. Available from: `https://doi.org/10.1145/3157079`.

Xin, Y., Kong, L., Liu, Z., Chen, Y., Li, Y., Zhu, H., Gao, M., Hou, H., and Wang, C., 2018. Machine learning and deep learning methods for cybersecurity. *Ieee access* [Online], 6, pp.35365–35381. Available from: `https://doi.org/10.1109/ACCESS.2018.2836950`.

Zhang, H., 2004. The optimality of naive bayes. *Aa*, 1(2), p.3.

Zhu, X.J., 2005. Semi-supervised learning literature survey.