



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Arjun Barrett; Rohan Bhowmik	Project Number J1401
Project Title Using Machine Learning and Infrared Imagery to Detect Drunk Drivers	
<p style="text-align: center;">Abstract</p> <p>Objectives Our goal was to develop an effective technique utilizing the power of artificial intelligence (AI) to prevent millions of alcohol-related crashes and save countless lives. We hypothesized that a convolutional neural network (CNN) could be developed and trained with infrared (IR) thermal images of the faces of drunk and sober individuals to estimate their blood alcohol content (BAC) within a margin of error of $\pm 0.01\%$.</p> <p>Methods We designed and built a convolutional neural network, a widely used AI technique that specializes in image recognition, then wrote a script to parse the images and train the network using an open-source database. The program is written in Python 3.6. It uses Tensorflow.Keras to train and test the model along and NumPy for data processing. We varied learning rates (0.00005 to 0.0002), epochs (15 to 25 iterations), and the number of images (1024 to 4096) to optimize the accuracy of the model.</p> <p>Results We found that the learning rate had the most significant impact on accuracy. The model's peak accuracy improves with increasing number of epochs and images. While the higher number of images and epochs, and lower learning rates improve the peak accuracy, these conditions lead to longer training times. Although the larger learning rates (0.0002-0.0001) yield higher accuracies (97.5% or $\pm 0.0043\%$ BAC), the network diverges with a larger number of training images. At lower learning rates (0.00065-0.00005), the models behave more predictably and show more consistent trends. However, due to slower convergence, the models with lower learning rates require more training images to hit peak accuracy, reaching 94.5% with 4096 images (or $\pm 0.0094\%$). If the model had been trained on more epochs and images, it would have achieved a peak accuracy greater than 97.5%. We also observed that the quality of the training data set has a significant influence on the end results, as an unbalanced ratio of sober to drunk images can lead to overfitting.</p> <p>Conclusions We developed an AI technique that can determine the blood alcohol content of an individual accurately with only an IR image. The consistently high accuracy of our neural network model proves that our hypothesis and exceeded our initial goal of achieving $\pm 0.01\%$ BAC margin of error. This technique can be developed into a smartphone app to provide accurate and fast results to replace the existing breathalyzers and blood tests used by law enforcement and doctors. Ultimately, automobile manufacturers could integrate IR cameras and our AI model in their cars to prevent drunk drivers from making it onto the road, saving countless lives.</p>	
Summary Statement Our project uses artificial intelligence powered by a CNN to accurately and instantaneously estimate the BAC of a person from an IR thermal image of the face, achieving 97% accuracy. Millions of lives could be saved by using this technique.	
Help Received We got help in understanding the effect of alcohol on body heat through an online research paper published by Dr. Hermosilla's group from the University of Santiago in Chile. We designed and tested the machine learning model and program by ourselves.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Yu-Ting Chang	Project Number J1402
Project Title Greedy, Yet Intelligent, Algorithms for the Game 2048 Using Python	
<p style="text-align: center;">Abstract</p> <p>Objectives The objective of this project is to design, implement, test, and analyze simple algorithms to find the one that can achieve the highest tiles in the Google Chrome Extension of the game 2048.</p> <p>Methods Materials/environment: iMac desktop computer, Python v3.5, Pycharm IDE, free open source packages OpenCV and PyAutoGUI, Google Chrome extensions 2048 by Moloko and Empty New Tab Page by thakis@chromium.org. Tested twelve self-programmed algorithms (including two baseline algorithms) 200 games each, outputting collected data to .csv files. Used Seaborn, Matplotlib, Pandas, and NumPy libraries to help me graph and analyze data.</p> <p>Results The optimal algorithms are a combination of chain reaction movement and the reward function given parameter 1 as the additional score (search depth 2 steps). Of those, the algorithm with the highest tile of the future board as parameter 2 appears to perform the best, and can reach the winning state of 2048 6.5% of the time.</p> <p>Conclusions I programmed and analyzed different algorithms for the game 2048. The optimal algorithm appears to be a combination of chain reaction movement and the reward function given parameter 1 as the additional score and parameter 2 as the highest tile of the future board. This demonstrates that a simple intelligent algorithm can achieve a good result comparable to that of more complex algorithms, given the right constraints and parameters to compare.</p>	
Summary Statement I devised a simple algorithm that can achieve the winning state of 2048.	
Help Received After learning techniques from internet searches and online courses, I programmed and tested the algorithm by myself.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Anusha Chatha; Arjun Chatha	Project Number J1403
Project Title Obstacle Avoiding Robot: A Reinforcement Learning Approach	
<p style="text-align: center;">Abstract</p> <p>Objectives Artificial intelligence is the wave of our future. We wanted to undertake a science project involving AI with real life applicability. Our goal was to program a robot that could detect and avoid obstacles using a Q learning algorithm. This capability could be applied in many helpful ways, such as in delivery systems or self-driving wheelchairs. We hypothesized that when the Q learning algorithm records more states, it learns an obstacle free path that covers a larger area of the enclosed space.</p> <p>Methods For this project, we obtained a LEGO Mindstorms Ev3 robot and attached an ultrasonic sensor to detect obstacles and a gyro sensor to make precise turns. We programmed the robot with a Q learning algorithm to avoid obstacles in an enclosed area by selectively moving forward or rotating to the right. In Q learning there are states, actions possible with each state, and a reward associated with an action. When the reward is high, the robot knows it performed the best action. The robot learns to choose moves that maximize the value of the reward. We first simulated the experiment on a computer using a virtual 11x11 coordinate plane.</p> <p>Results We performed over 300 trials with different parameters to make the Q learning algorithm run successfully. We tested the influence of various parameters on our algorithm and determined that a discount factor less than 0.75 makes the robot learn in fewer than thirty moves and the probability of random moves must be less than 0.1 for the algorithm to learn in a consistent manner. The effectiveness of the algorithm is influenced by how the states are encoded. Our state function remembers the last n moves made by the robot. We demonstrated that the area of the enclosed space covered by the robot increased as n was increased from 2 to 10.</p> <p>Conclusions Our experiments confirmed the hypothesis, that increasing the number of states recorded by the robot enabled it to cover a larger area. Once we programmed the Ev3 robot with our Q learning algorithm, it was remarkable to see it adapt to new obstacles and surroundings as the algorithm learned and taught itself.</p>	
Summary Statement We devised a Q learning algorithm that enabled a robot to learn a repetitive path within an enclosed space which avoided all obstacles.	
Help Received We received help from our parents to write and debug the Q learning based simulator program in python. We modified the simulator program for each experiment on our own. Further, we also modified the program for moving the robot by including functions for actuating the motors and reading the sensors.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Mason Choey	Project Number J1404
Project Title Finding the Winning Strategy for Playing Gin Rummy: A Computer Simulation	
<p style="text-align: center;">Abstract</p> <p>Objectives The objective of this study is to determine the gin rummy playing strategy with the highest likelihood of winning.</p> <p>Methods Computer, Python coding language. Coded a 2-player gin rummy game in which the computer plays itself, using different strategies to test and measure the effectiveness of each strategy. The playing strategies tested were: (1) Highest Card Discard (player discards highest cards in their hand), (2) Lowest Card Discard (player discards lowest cards in their hand), (3) Random (player discards any cards not part of a meld and draws randomly from the draw pile and discard pile), and (4) Learning Strategy (computer randomly picks one of the 3 strategies to play and if the player wins, continues to play that strategy for the next game, otherwise it switches strategies).</p> <p>Results After the computer simulates 10,000 two-player games for each match-up, the wins, ties and losses are tallied. The Lowest Card Discard strategy won the majority of games against all of the other strategies with the highest margins against Random strategy (83% vs 17% wins) and significant advantage over Highest Card Discard strategy (63% vs 36% wins) and Learning Strategy (65% vs. 35% wins). For the greatest probability of winning, the computer decides to play Lowest Card Discard strategy.</p> <p>Conclusions Although a commonly acknowledged winning strategy for gin rummy is Highest Card Discard (to minimize the number of high cards deductions from the total score), my program proved that the opposite strategy, Lowest Card Discard, results in a higher chance of winning. In other words, players should focus on making melds with high cards in order to maximize their total score. My program is able to test each strategy against another, in isolation, and complete 10,000 game simulations in less than 5 minutes. To replicate these results, it could take several months for people to play with regular cards, for 24 hours a day, which is not realistic. Computer simulations prove to be an effective and efficient way to test playing strategies and pinpoint the strategy with the highest chance of winning. To refine my study further, my next steps would be to add the ability to create runs (or sequential melds), add more strategies to be tested, and add the ability to change strategies in the middle of a game.</p>	
Summary Statement My computer simulations prove that discarding the lowest cards in your hand is the strategy with the highest likelihood of winning at gin rummy.	
Help Received My mentor, Nathan Hutchison, a Computer Science graduate student at Santa Clara University, has been teaching me advanced Python programming and helped troubleshoot bugs over the 8 months that I worked on the program. This project, however, was entirely my idea and I wrote all of the code.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Eitan Feldman	Project Number J1405
Project Title Spot That Dot: Autonomous Classification of Benign and Malignant Moles	
<p style="text-align: center;">Abstract</p> <p>Objectives The objective is to create software to identify malignant moles by evaluating size, shape, and variation in color. This software attempts to compensate for changes in distance and camera orientation.</p> <p>Methods Computer, camera, circular sticker Tested mole classification software algorithm on internet images of benign and malignant moles as well as images taken at home of moles with adjacent sticker</p> <p>Results Statistics on the accuracy of the mole classification software revealed that positive and negative predictive values, sensitivity, specificity, and accuracy were 0.8. After correcting distortion in images, the average deviation from the mean of mole to sticker ratio was 45.2%, 46.8% and 52.5% for each of three moles.</p> <p>One aspect of the hypothesis was found to be untrue: camera orientation must be controlled so that the circular sticker has an aspect ratio of less than 1.2. Otherwise the software was unable to produce consistent estimates of the mole size (relative to sticker size). The other aspect of the hypothesis, that software can evaluate characteristics of a mole to determine whether it is cancerous, was largely proven.</p> <p>Conclusions The 80% percent diagnostic accuracy achieved with this software compares favorably with trained professionals having five or fewer years of experience. The software also provided evidence that shape irregularity and variation in color are important indicators of melanoma.</p>	
Summary Statement This project identifies moles and classifies them as benign or malignant based on size, color variation, and shape, but was not able to correct for large variation in camera distances and angles.	
Help Received Mother helped design poster. Father provided coding advice.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Gabriel Fergesen	Project Number J1406
Project Title Using Artificial Intelligence and Electroencephalography against Autism and Attention Deficit Disorder	
<p style="text-align: center;">Abstract</p> <p>Objectives Can an Artificial Intelligence with electroencephalography predict when the user is unfocused. What is the best electroencephalography headset to utilize for the experiment? Which software to use, and what to avoid. How to access the programs I created. How to configure Wekinator and Muse Monitor. How to work with OSC data. How to train an artificial intelligence in Wekinator and run an experiment. How to approach getting a mentor, and attempt to secure product donations</p> <p>Methods The procedure I followed was a system of three phases.</p> <p>The first phase is to develop and procure the tools, computer programs, artificial Intelligence and expert mentors in the areas of computer science, computer programming and data analysis.</p> <p>I was not able to complete this stage because I could not reconcile the headsets to the Artificial Intelligence (A.I.). Consequently, I could not run the trial portion of this experiment. To complete this I believe I will need more experience or experts.</p> <p>The second phase will be running the experiment when I get the individual components online.</p> <p>The third phase will be assembling a product which will be, a standalone headset, a stimulus to wake the user and taking the product to Kickstarter, a platform for crowdfunding an idea and bringing it to marketplace.</p> <p>Results Since I was unable to run my experiment my results were inconclusive, but this is my best educated guess on how to manage the data without being able to conduct the experiment. Repeat experiment 3 or more times. Put the data into a chart that shows the human guess and the bot's guess which will prove whether it's possible to prevent the mental state of spacing out via AI augmented shock therapy from intelligent analysis of EEG.</p>	
Summary Statement Gather the tools, programs, artificial intelligence and expert mentors in computer science, programming and analysis, Run experiments & complete a standalone headset with the ability to wake the user and take the product to the marketplace.	
Help Received Natalie Bloome, Expert, Github Dr. Rebecca Fiebrink, Computer Expert, overcome errors Patricia Tsori-a-sue, Inventor of Wekinator, create a map and timeline Brian Cooper, UC Long Beach Manager, hours debugging code Dr. Luciano Nocera, PATA Science Institute, USC - Numerous tasks	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) David Freifeld	Project Number J1407
Project Title Detecting Exoplanets with Anomaly Detection and Hotelling's Theory	
<p style="text-align: center;">Abstract</p> <p>Objectives The objective is to determine which anomaly detection method is the most accurate at classifying lightcurves of stars, and which is the most suitable for use by astrophysicists.</p> <p>Methods A computer with Python 2.7 and multiple machine learning modules installed, and three base open-source machine learning programs were used. These were modified and tested for accuracy on labelled lightcurve data. I used cleaned lightcurve data from the Exoplanet Hunting in Deep Space challenge on Kaggle.</p> <p>Results Modified three programs to utilize anomaly detection strategies to classify lightcurves, and were tested on a labelled dataset. Many tests were run with different input parameters to determine the best performance of each strategy. Hotelling's Theory proved to be the best, having the highest accuracy of 74.7% when used with a threshold of 0.3, a dip threshold of 175, and a cutoff of 1500.</p> <p>Conclusions Hotelling's Theory ran significantly faster than the other two programs and was the most accurate showing that it was a effective method for detecting exoplanets. It's speed shows that it is suitable for use on large amounts of data, and these two features allow a significantly higher amount of exoplanets to be discovered in a shorter timeframe than the traditional manual method.</p>	
Summary Statement I found that Hotelling's Theory is a suitable method for detecting exoplanets automatically as shown by high testing accuracies and fast runtime.	
Help Received Andrew Lookingbill advised me on how to set up LSTMs and RNNs, along with Conner Vercellino who advised me to begin the project focusing on the dataset.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Allison Hung	Project Number J1408
Project Title Iterated Prisoner's Dilemma	
<p style="text-align: center;">Abstract</p> <p>Objectives The objective of my project is to compare several different strategies used in the game Iterated Prisoner's Dilemma. I investigate how each strategy performs against all other available strategies, and which strategies are most/least successful overall.</p> <p>Methods For this experiment, I wrote a program in Python that implemented 11 different Iterated Prisoner's Dilemma strategies, and had them compete against each other in a round-robin tournament. In each matchup, a pair of strategies played against each other for 200 iterations. For each iteration, the following scoring system was used: (a) if both strategies cooperate, they receive 3 points each; (b) if both strategies defect, they receive 1 point each; (c) if one strategy cooperates and the other defects, they receive scores of 0 and 5 points, respectively. Each strategy competed against every other available strategy, including itself, for a total of 121 matchups. The entire tournament was run a total of 10 times, to account for the random factors present in some of the strategies. Scores per iteration were averaged across all tournaments.</p> <p>The strategies used were: Always Cooperate (AC), Always Defect (AD), Tit for Tat (TFT), Massive Retaliatory Strike (MRS), Random (R), Soft Majority (SM), Firm but Fair (FBF), Naïve Prober (NP), Tit for Two Tats (TFTT), Two Tits for Tat (TTFT), and Pavlov (PV).</p> <p>Results From most to least successful, the strategies were: SM (average 2.68 points per iteration), TFTT (2.66 pts), MRS (2.64 pts), FBF (2.62 pts), TTFT (2.61 pts), PV (2.61 pts), TFT (2.59 pts), AC (2.56 pts), R (2.17 pts), NP (1.94 pts), and AD (1.92 pts).</p> <p>Conclusions By using only 11 strategies in my experiment, I was able to analyze each strategy's performance carefully, and understand why certain strategies outperformed others. My hypothesis, based on background research, was that the best strategy would be TFT. However, SM outperformed TFT, mainly because SM did better against NP than TFT. Similarly, I predicted that the worst strategy would be AC, but because many of the strategies used in this experiment rewarded cooperation and punished constant defecting, AD ended up losing instead of AC.</p> <p>It's important to note that my results were specific to these particular strategies. Every strategy used impacts the performance of every other strategy. Therefore, if different or additional strategies were used, the final results would be different.</p>	
Summary Statement I implemented an Iterated Prisoner's Dilemma tournament in Python and analyzed the performance of 11 different strategies that competed against each other.	
Help Received My mom taught me about classes and objects in Python. My dad taught me about standard deviation and standard error.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Alan Lee	Project Number J1409
Project Title Machine Learning for Distinguishing Benign from Malignant Liver Tumors	
<p style="text-align: center;">Abstract</p> <p>Objectives Liver cancer is the second deadliest cancer in the world. Due to the multitude of lesions found within the liver, locating a specific lesion and determining its malignancy is a challenging yet urgent task. The objective of this study was to investigate machine learning's ability to distinguish benign and malignant liver tumors and to evaluate how effective such differentiation would be.</p> <p>Methods A total of 838 liver tumor MRI scan images from published papers were collected and grouped into the four scan types T1 No Contrast, T1 Arterial, T1 Interstitial, and T2. Next, I derived an additional 634 cropped images by focusing on the lesion area. The 1472 images thus obtained were used to train and test a convolutional neural network (CNN) model implemented with Python Keras library. 80% of the images were randomly selected for training and the remaining 20% for testing. A confusion matrix was used to measure the accuracy of the model.</p> <p>Results Training the cropped images in combination with the original images yielded accuracies of 81% for T1 Arterial, 78% for T1 No Contrast, 79% for T1 Interstitial, and 85% for T2, a significant improvement from when only original images were used for training. Testing all 1472 images together regardless of the scan type was 81% accurate in predicting whether the lesion was malignant or benign.</p> <p>Conclusions This study showed that machine learning (ML) model could successfully distinguish malignant and benign liver tumors. Cropping images to give the ML model a focus point around the lesion area improved the decision accuracy. Testing the four scan type groups individually also provided a quadruple cross-check on results and precision. Future studies will involve training with larger quantities of data and the creation of synthetic images to further increase accuracy.</p>	
Summary Statement I analyzed if machine learning could be effective at distinguishing benign and malignant liver tumors.	
Help Received I received help from my computer science teacher, Jared Lera, who introduced Convolutional Neural Network to me. Special thanks to my mother who provided guidance and online library access to images.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Houjun Liu	Project Number J1410
Project Title Observing Behavior of Actor-Critic Driven Disturbance Agents on Cryptocurrency Exchanges	
<p style="text-align: center;">Abstract</p> <p>Objectives In this study, an example of how automated agents can potentially swing a cryptocurrency trading environment is presented. Using a group of simulated markets, reinforcement learning agents will be trained to negatively affect the asset's price. Simulated markets are created and populated by two classes of agents -- the trader and the disturbance -- one trained to mimic the behavior of regular users from a target trading environment (in this case, the Poloniex Exchange), the other trained to potentially disturb the former class of agents' decisions and drive the market price down.</p> <p>Methods First, a group of agents (the trader) is trained simply to emulate the patrons of the real market -- a core component in building the training environment. These simple trading agents are an 8-layer-long Deep Neural Network that takes market data as input and generates a scalar that represents an order placed. During testing, downloaded market data and order history are split 90% to 10% for training and validation respectively. With a trained trader network, multiple instances of the graph are launched -- each acting as a separate patron in a collective network. Separate instances of such trained graphs will interact with each other within an emulated exchange, which serves as the environment for the disturbance agent. Next, the disturbance agent trains with the environment as highlighted above. The policy network (actor) takes input from the market data and creates an output scalar representing an order -- similar to the trader agent; the advantage value network (critic) is trained based on rewards assigned by the amount of reduction in the environment market's price. As the agent reaches a target low price, it is considered a terminal state with a reward value of 1. In a similar fashion, should the agent lose all funding provided, it is also considered a terminal state; however, a reward of -1 is assigned in this case. All other rewards are assigned based on the negative change in market price of the environment normalized between 0 and 1.</p> <p>Results During 10 trials, each one with 15 minutes of data collection (resulting in about 45,000-50,000 data items per trial), 9 separate distributions of trader and disturbance is tested -- increasing the amount of disturbance by 5 agents at one time. The price change (Price Delta) is calculated by dividing the change in price before and after the attack by the initial market price. The average price delta showed a decrease in market price of market price starting at a 15:35 disturbance to agent ratio; finally reaching the hypothesized 10% decrease in market price at a 45:5 disturbance to agent ratio.</p>	
Summary Statement Algorithmic agents can easily influence the trend of a cryptocurrency exchange if a market's volume is relatively low.	
Help Received Mr. Heiko Ritter reviewed the scientific process, literature, and my work to guarantee this project's integrity. Mr. David Babington and Ms. Susan Cole helped verify that my usage of English syntax was accurate.	



**CALIFORNIA SCIENCE & ENGINEERING FAIR
2019 PROJECT SUMMARY**

Name(s) Keshav Narang	Project Number J1411
Project Title Generalizing a Solution to the Tower of Hanoi with Varying Number of Pegs and Rings	
<p style="text-align: center;">Abstract</p> <p>Objectives The Tower of Hanoi is a problem where in a system of m pegs, n rings are on the first peg, arranged from smallest on the top to largest on the bottom. One ring can be moved at a time, and a larger ring can never be on top of a smaller one. The objective is to find the minimum number of moves needed to move all n rings from the first peg to the last one.</p> <p>The Tower of Hanoi problem is valuable for it can show the most efficient way to solve problems, and it is connected to important concepts, especially mathematics, as it is related to Pascal's Triangle, used for binomial expansion, and Sierpinski's Triangle, a fractal.</p> <p>My evaluating criteria for this project was to find a general formula to calculate the minimum number of moves needed to complete the Tower of Hanoi given the number of rings and pegs present in the system.</p> <p>Methods 1. Using http://towersofhanoi.info/Animate.aspx, make a list for the minimum number of moves needed to transfer n rings for m pegs where n is an integer in the range 1-40 and m is an integer in 3-8. 2. Observe the patterns in the data, particularly how many times the minimum number of moves increases by 2 for each peg, and then how many times it increases by 4 for each peg, and so on. 3. Note that this pattern follows Pascal's Triangle, and use this fact to come up with a formula for the Tower of Hanoi.</p> <p>Results My results showed that increasing the number of pegs decreases the number of minimum moves required to complete the game. A pattern that all the pegs followed was that as the number of rings increase, the minimum number of moves increased by one (2^0) move for a number of times, then started increasing by two (2^1) for another number of times, then increased by four (2^2) moves, and so on. After analyzing the number of times the moves increased by one for each peg, and then two moves, and so on, Pascal's Triangle became visible as a pattern throughout.</p> <p>Conclusions In conclusion, I did find a formula for the Tower of Hanoi through Pascal's Triangle and combinatorics, which can be used to make a relatively simple equation.</p>	
Summary Statement I Generalized a Solution to the Tower of Hanoi with Varying Number of Pegs and Rings	
Help Received Ms. Jiu Chang	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Neel Redkar	Project Number J1412
Project Title Using Machine Learning to Identify and Sort Waste	
<p style="text-align: center;">Abstract</p> <p>Objectives The goal for this project is to design, build, and test a mobile application (app) that is able to sort waste into recycling, compost, and trash. Only 9% of recyclables gets recycled a large percentage, 79%, ends up in landfills. Another goal of the project was to allow the app to further sub-classify into types of recycling - paper, plastic, metal, glass, and cardboard.</p> <p>Methods The project required image recognition with machine learning to build my project to allow detection and sorting of the image of the waste. I achieved this by using CoreML machine learning from Apple, and Swift/ XCode from Apple to build the application. I built all of the software including the image recognition model and the full application including the user interface. The procurement of 3600 images and training was also done my me. In the testing phase of the product, I procured about 70 pieces of waste, tested the app, calculated and plotted the accuracy rates.</p> <p>Results The user interface of the app was intuitive; users were able to understand the intent of the app (RecycleSmart) and how to use the app. The accuracy rate for the ML model was 91.4% for the 70 pieces of waste that were given for a variety of each categories, 10 each.</p> <p>Conclusions The app was able to sort the waste with a 91% accuracy rate, which is adequate to use in people s everyday life. It was also easy to use, with the user interface doing well in usability. This means that people would be able to download the app from an Appstore and be able to use it with relative ease.</p>	
Summary Statement I created a app to sort waste into recyclable, compost, and trash to educate and help people to put waste in the correct bin.	
Help Received None. I created, tested, and deployed the app myself.	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Sanskriti Singh	Project Number J1413
Project Title Detection of Pneumonia on CXR Image Using Convolutional Neural Networks with an Emphasis on Minimizing False Negatives	
<p style="text-align: center;">Abstract</p> <p>Objectives The objective of this project was to develop an efficient algorithm to detect pneumonia looking at a Chest Radiograph X-ray image with an emphasis on minimizing False Negatives. World Health Organization states, "Pneumonia is the single largest infectious cause of death in children worldwide". It has been shown that convolutional neural networks (CNN) are good at extracting features from images. After finding the best CNN architecture, I wanted to further minimize the False Negatives without sacrificing the quality of the model because the False Negatives have a higher risk of being ignored. I did this by taking a random portion of the False Positives during training, changed the label to True and retrained the model. Then I took the best models and ensembled them for the final prediction on the unaltered test set.</p> <p>Methods The RSNA dataset is an annotated dataset with labels 0 (no pneumonia) and 1 (with pneumonia). The NIH is another annotated dataset with 15 different classes, one of which is pneumonia. I augmented the RSNA dataset by flipping the true labeled images horizontally, vertically, and 50% of the time both to make it balanced. To manage compute resources, each datasets' images were resized into 256 by 256. The RSNA dataset was then split into a train and a test set of 80:20%. The training dataset was further split into 80:20% as train and validation set. To find the best model architecture, I ran multiple experiments with a different number of channels and layers. The experiments started with simple architectures and slowly started graduating in the number of layers and channels. I used the ensembling technique of Max Voting to further improve the prediction result. The quality of the model is measured by the F1 metric. To minimize False Negatives, I changed the target of a portion of the False Positives to be True in the training dataset and retrained the models.</p> <p>Results I received an F1 score of 0.58, a precision of 0.6, recall of 0.56, and AUROC of 0.73, without an emphasis on the minimization of False Negatives. With the technique of minimizing False Negatives, I was able to increase the recall by 12% while keeping the same F1 score and the AUROC also increased by 3%. As a comparable, Stanford's model, ChexNet (2018), has an AUROC of 0.76 in the detection of pneumonia on the NIH dataset.</p> <p>Conclusions I constructed the best CNN architecture after systematically training various models with different numbers of layers and channels. I further improved the recall of the best trained models by a proposed technique.</p>	
Summary Statement I developed and proposed a novel technique to increase the recall in detection of pneumonia on CXR image using CNN, while maintaining the F1 score.	
Help Received I designed and trained the CNN models by writing program in python using Keras framework with the help from internet search and guidance from Mr. Manish Singh (Principal Engineer @ Ambarella). I learned about the disease and discussed the results with Dr. Parvathy (Bayside Medical Group, CA).	



CALIFORNIA SCIENCE & ENGINEERING FAIR 2019 PROJECT SUMMARY

Name(s) Vibha Srinivas	Project Number J1414
Project Title Automatic Invasive Insect Identifier with Alerts Using Artificial Intelligence	
<p style="text-align: center;">Abstract</p> <p>Objectives The two main goals of my project were to reliably detect invasive insects and design a platform to alert the presence of invasive bugs to farmers and to state and local agricultural departments.</p> <p>Methods An Image-Classifer, a type of AI neural network, was trained on 25 images each of 5 invasive insects and 4 insects native to California. A station built with a Raspberry Pi, camera, and light attraction mechanism was used to attract insects and notify farmers via SMS if the insect was invasive or not.</p> <p>Results After being trained with the images, the station was given an independent testing set of insect images, for which it could detect whether the insect was invasive or not with an accuracy of over 70%.</p> <p>Conclusions Because the software has such a high success rate in detecting invasive species, this method would be a much faster way of eliminating these bugs from agricultural fields. Because these stations can be easily built and spread across an area, we can limit the ecological and agricultural impact that invasive species could have.</p>	
Summary Statement I built an insect-detecting station that uses AI to automatically alert farmers and local agricultural departments when invasive bugs are in their farm fields.	
Help Received I designed, built, and tested the station myself. I used Google's Tensorflow framework to train my images.	