Face Mask Recognition

Emma Ascolese, Theodore Donegan, Stephanie Keene, Brennen Hogan

Abstract

Both the Centers for Disease Control and the World Health Organization currently recommend masks for the general public, but earlier in the pandemic, both organizations did not recommend mask wearing. The change in guidelines has caused confusion among the public regarding the utility of masks. As people return to their normal lives, mask wearing becomes increasingly important for maintaining public health and preventing the spread of disease. Namely, mask wearing is particularly effective when going to stores, attending classes, visiting large gathering spaces, or using public transportation. In our project, we leveraged a facial image dataset with over 5,900 unique entries and 3 classifications. The classifications we used are "face_with_mask", "face_no_mask," "face_with_mask_incorrect." Our team has worked to develop two neural networks which classify facial images and determine whether the pictured individual is wearing a mask, not wearing a mask, or wearing a mask improperly. Our first model is a traditional neural network which uses 2D convolution layers and max pooling, and our second model is a pre-trained resnet deep neural network which we trained on our mask dataset. Both have proven to be proficient at classifying facial images into the three categories outlined above. These systems could be used by stores, public transportation centers, or schools to mandate the proper use of masks when in public settings. To mitigate the spread of disease, specifically COVID-19, we hope to increase adherence to the mask wearing protocols outlined by the CDC throughout the general public ("How to Select, Wear, and Clean Your Mask"). The use of masks in a public setting has also been proven to generate social good. The ethical frameworks of feminist or care ethics, deontological ethics, and utilitarian ethics support the notion of increased mask usage. The use of an intelligent machine to identify proper mask wearing will enforce standard guidelines recommended by the CDC and WHO, generate social good through the use of computer science and AI, and encourage behaviors that lead to an overall healthier society.

1. Introduction

Problem Statement

Amidst the COVID-19 pandemic, wearing face coverings is of paramount importance to protect our communities and slow the spread of the virus. As businesses begin to open their doors, owners seek to keep patrons and staff members safe through mandates stating that people within the store must be wearing masks correctly. Despite the clear postings outside establishments, some people do not comply. The problem being addressed is the lack of emphasis on mask wearing in public spaces. Specifically, we are attempting to use AI as a means of enforcing the CDC's face mask protocols. We are accomplishing this through developing two convolutional neural networks which can identify improper mask usage and can help to encourage the adherence to CDC best practices regarding face coverings and masks. This includes three-classifications: not wearing a mask, wearing a mask, wearing a mask incorrectly. Violations of public health standards can put employees in awkward confrontations, or could lead to the spread of disease. This project aims to take the human out of the confrontation and prevent people from entering public spaces without masks.

We are looking to develop technology that could serve as a mask detection system and could be placed at store fronts to identify people who are incorrectly wearing masks. This system would notify the violator that they are not complying with the space's policies, and would remind them to adhere to the CDC standards before entering the building. The technology protects other customers from being exposed to people who are not wearing masks and those who are wearing them incorrectly. It also protects the employees from exposure and awkward (potentially violent) confrontations with people who knowingly violate mask mandates. This project promotes the virtue of diligence and supports societal good by reducing the opportunity to spread the disease. If similar technology were to be implemented in the public sector, the adherence to CDC protocols would rise and the rate of disease transmission would fall. We aim to develop a system able to identify the presence and the correct application of a face mask in order to create a safer environment in public spaces through recognition and notification of absences or misuses of masks with a machine.

Ethical frameworks

The ethical and social need for face mask recognition is falls under multiple frameworks: feminist/care, deontological, and utilitarian ethics.

Feminist ethics is considered the "ethics of care." It is an ethical framework which encourages protecting the most vulnerable individuals in a population. In the case of COVID-19, the most vulnerable would be those who are immunocompromised, elderly, or lack sufficient healthcare. Enforcing mask-wearing in public spaces protects these vulnerable groups the most, as they could face severe consequences if they were to contract the virus. All three of these groups are at risk for serious complications upon contracting the virus, so any attempt to decrease the spread will be beneficial for them. Furthermore, feminist ethics encourages selfless acts. In terms of protecting others from disease, wearing a mask is one of the most effective means of prevention. Thus, this behavior could be considered selfless. (Norlock)

Deontological ethics revolves around following one's duty to fellow human beings. Deontology stands in opposition to consequentialism and other moral theories which focus on the results of an action. The categorical imperative is a central guiding principle of deontology. It states that one should always act in such a way that the maxim of one's action can be universalized. A maxim is oftentimes referred to as the agent's intention in a situation. The categorical imperative and the concept of universalizability are particularly useful when approaching the topic of mask wearing due to effectiveness of the widespread adoption of the behavior. If everyone were to wear masks and the action was universalized, mask wearing would reach maximum effectiveness. Encouraging mask wearing is adamantly favored by deontological ethics for this reason. (Alexander)

Utilitarian ethics seeks to produce the greatest amount of good for the greatest number of people, resulting in the highest net positivity. Since wearing a mask is known to protect those around the mask wearer and stop the spread of COVID-19, properly wearing a mask is the best way to put an end to the pandemic and reduce the number of infections and deaths. Wearing a mask does little to no harm, excluding those with health conditions, so the most good for society is done when everyone wears a mask. Therefore, our project is ethically sound from the utilitarian perspective. (Driver)

2. **Related Work**

After making masks mandatory for riders and drivers, Uber developed a mask recognition protocol using the front-facing camera on smartphones. The mask recognition step is not mandatory for all riders and drivers. It is required after any rider or driver has been flagged for not wearing a mask during a past ride. Similar to our project, this technology prevents the drivers from needing to constantly remind uncooperative riders. Uber's algorithm does not use any facial recognition technologies, instead it detects the mask as an object on the face (Kansal). Our proposed approach is different from Uber's, but the viability of another tactic is worth noting. This is proof of a similar automated system being used effectively in the public transportation sector and demonstrates the usefulness of the concept.

Motorola Solutions created a No Face Mask Detection video-based technology that enables real-time responses. The technology uses AI-enabled analytics to detect objects in the camera's view. It first detects if there is a human, and then looks to analyze whether the person is wearing a face mask. When a violation is detected, an alert system goes off ("No Face Mask Detection"). A system similar to the one used by Motorola could be used in various public settings including stores or gathering spaces. Motorola's solution is unique, as it analyzes images in real-time. Though our proposed solution takes static images, real-time response could be a useful application of the technology going forward.

IEEE has studied and proven the effectiveness of convolutional neural networks being used for facial recognition. They were able to achieve over 95% accuracy through classification using an eigenface algorithm (Jamil). Facial recognition has been widely studied and used across the industry, and has been proven to be commercially viable in products such as the iPhone ("An On-device Deep Neural Network for Face Detection"). A deep neural network, similar to the one mentioned in the Apple article, is used in our resnet34 model. Deep neural networks prove to be a promising method for learning low-level features of an image. Convolutional neural networks have been proven to be an effective means of analyzing facial data, and this application of AI should be appropriate for our project. Our proposed approach involves the use of a 2D convolutional neural network for our custom model.

There is another project that is similar to this one, done by Tryolabs. Their project uses computer vision and artificial intelligence to conclude if patrons are wearing their masks in a video stream, which would be suitable for use in stores and other public places. The Tryolabs team used facial detection from a video stream to recognize the faces to feed to the model. This is similar to this project, except the images in our project come from a data set of photos as opposed to a video stream. Then, the Tryolabs team used a resnet neural network to train a model to classify the images found, very similar to our proposed approach. The successful application of a resnet model to identify mask usage supports our proposed resnet model. (Tryolabs)

3. **Approach**

We worked with image data from a Face Mask Detection Dataset on Kaggle ("Face Mask Detection Dataset"). This dataset contained over 15,000 entries and 20 unique labels. Images in the dataset can also contain multiple face images with different labels. In order to account for this, we needed to augment the data before use. First, we ran a script to only use images that had the labels "face_with_mask", "face_no_mask", "face_with_mask_incorrect." This trimmed the dataset down to 5,900 entries and 3 unique labels. In both the custom model and the resnet model, we read through the augmented dataset and created an image standardizer class which preprocessed the images before they were turned into tensors. Each facial image has coordinates in the picture, so we first cropped the image to just contain the face. Next, we scaled all images to a standard size of 224 by 224 with a transform function. From that point onward, we could guarantee that all images only contained faces and had a standard size. This allowed us to eliminate all non-facial images from the dataset and mitigate the randomness due to our data. We used a python DataLoader from torch.utils with a batch size of 10 to take the image data from the class and run it in the training and testing phases. For testing purposes, our model used 75% of the data for the training set and 25% of the data for testing.

For the two-classification problem, we used softmax as the post-processing function for both models. This function transforms values to either 0 or 1, which was ideal in the case of the two-classification problem. The softmax function serves as a rounding function with a normalized distribution (Wood). In the case of the three-classification problem, we used TopK as our post-processing function for both models. For a three-classification problem, we needed to use probabilistic post-processing. Within our three-classifications (mask, no mask, and mask worn incorrectly), the TopK function will select the classification that has the highest probability based on a given input. The results in Appendix B indicate that this decision proved to be an effective method of classification. We also utilized the cross-entropy loss function as our loss function for weight adjustments in both models. Cross-entropy is the default loss function to use for multi-class classification problems (Pytorch.org). In this case, it is intended for use with multi-class classification where the target values are in the set {0, 1, 2}, where each class is assigned a unique integer value.

For our 3-classification custom model, we used two 2D convolution layers with a nonlinearity (ReLU), and max pooling. For image data, 2D convolutional layers prove effective at creating a feature map. This accounts for not only the presence of a feature, but also the location of that feature in the image. Since the image data was standardized, and only facial data was used, we could guarantee that the masks would be in a similar location in each image. We used multiple convolution layers in our model, since layering these allows for each 2D convolution to learn specific features. The deeper the layer is, the more abstract of a feature it can learn. A limitation of the feature map output of convolutional layers is that they record the precise position of features in the input. Thus, slight movements in the position of the feature in the input image will result in a different feature map. This would be considered an instance of "overfitting" a model to a certain type of image. A common technique for dealing with this overfitting problem is down sampling through pooling. Traditionally, a pooling layer is added after a nonlinearity (ReLU) has been applied to the feature maps. Our custom network uses two sets of the 2D convolutions, ReLU, and max pooling layers. (Brownlee)

Resnet or residual network learning is a type of deep convolutional neural network. Since our custom model was a traditional convolutional network, we wanted to compare our results to a more advanced variant which uses a similar technique. The number of layers in a deep neural network enriches the number of low, middle, and high-level features and classifiers for a given image. Resnet34 is the shallowest of the class of pretrained resnet models, and we selected this one since it would be the most similar to our custom neural network. A limitation of our custom model is the number of layers, and the resnet34 model differs by offering the capabilities of a deep neural network. The additional layers in the resnet model should enable it to learn more low-level features and could lead to improved accuracy. Resnet34 is also a classification model made for images, as it is pretrained on the ImageNet dataset. This dataset features over 100,000 unique images. Having been exposed to additional training before the training on our data set, and being equipped with more layers than our custom model, we expected resnet to outperform our custom neural network. (He)

4. **Results**

We ran both the resnet and custom models on two different datasets. The first dataset, which is found in Appendix A, features only two-classifications. The models must distinguish between mask and no mask in this case. In Appendix B, data from our three-classification problem can be found. This features data for masks, no masks, and masks worn incorrectly. Both models were run for learning rates of 0.003, 0.005, 0.007, and 0.009 to also compare how changing the learning rate would impact the results.

Based on the data in **Tables 1.1 and 1.2**, the resnet34 pretrained model outperformed the custom neural network in the two-classification test runs. The average testing loss of the resnet model was only 0.0063115 compared to an average loss of 0.03269375 for the custom neural network. This indicates that the average result before post-processing was more accurate for resnet than the custom model. This evidence is also supported by the data in **Table 1.2** which shows that the resnet model averaged 1405.5 (97.74%) correct predictions out of 1438 possible entries for all learning rates compared to an average of just 1271.5 (88.42%) for the custom model. This difference of almost 10% indicates that the resnet model is especially effective after the post-processing. Both models demonstrated a high level proficiency at identifying faces with and without masks.

For the three-classification problem, the resnet34 model also outperformed the custom neural network. Based on the data in **Tables 2.1**, **2.2**, and **2.3**, the resnet34 pretrained model was proven to be more effective for all learning rates and all success metrics. **Table 2.1** shows that the resnet34 model's average testing loss was 0.015609757 compared to the average testing loss of 0.04231625 for the custom neural network. This demonstrates that on average, the resnet34 model was closer to the target when calculating loss using the cross-entropy function. As demonstrated in the two-classification data, a low testing loss is a strong predictor of post-processed success.

Results in **Table 2.2** and **Table 2.3** are based on the post-processed predictions given by the neural network. For the 3 classification models, we used the TopK function to select the class with the highest probability of correctness based on the data. A prediction was counted as correct in **Table 2.2** if the model correctly predicted the class which the image belonged to out of face_with_mask (0), face_no_mask (1), face_with_mask_incorrect (2). In **Table 2.3**, we scored the models based on the effective correct predictions. In this case, we counted a prediction as correct if it was an exact match, or if the model predicted either face_no_mask (1) or face_with_mask_incorrect (2) for a target that was either 1 or 2. We created this metric due to observed behavior in model training. Oftentimes, the face_with_mask_incorrect (2) category was confused for face_no_mask (1). When considering the application of this technology in a real world setting, both cases 1 and 2 would produce the same result from a customer perspective. The machine would alert the customer of the CDC violation, and require them to wear a mask prior to entering. Comparing **Table 2.2** and **Table 2.3** demonstrates that the model does misclassify categories 1 and 2, but results are oftentimes highly accurate regardless of the metric. Out of a total 1475 possible test entries, the average number of correct predictions for all learning rates was 1266.25 (85.84%) for the custom model and 1420 (96.26%) for the resnet34 network. Similarly, the average number of effective correct predictions for all learning rates was 1301.25 (88.22%) for the custom network and was 1442.25 (97.78%) for the resnet34 model. In both cases, the resnet34 CNN was better than the custom network by about 10%. The custom network saw a greater improvement in accuracy due to the effective correct metric. Both the actual correct predictions and effective correct predictions demonstrate that the resnet34 model was more accurate than the custom network when using the TopK post-processing function. The correct prediction percentage for this iteration of the models was slightly lower than the correct prediction percentage for the first two-classification models. The data for the initial two-classification models can be seen in **Table 1.1** and **Table 1.2**. This decrease in the average was around 3% for the custom model and 1% for the resnet34 model. This is expected behavior for adding an additional classification.

As for the most effective learning rate, the custom network had the lowest average testing loss with the rates of 0.003 and 0.007; however, the learning rate of 0.003 yielded the most accurate results in **Table 2.2** and **Table 2.3**. For the resnet34 model, the learning rates of 0.005 and 0.007 had the lowest average testing loss. In both **Table 2.2** and **Table 2.3**, the learning rate of 0.007 narrowly outperformed the rate of 0.005. Overall, learning rates did not have a dramatic impact on the model's accuracy, but the learning rate of 0.003 seemed to be the best for the custom model and the learning rate of 0.007 produced the best results for resnet34.

The pretrained resnet34's performance demonstrated the effectiveness of deep neural networks for facial recognition and image data. Furthermore, this model's success demonstrates that prior training, even if it did not include any mask data, could improve the accuracy of a neural network.

5. **Conclusion**

Reflection on experience

This project gave us experience in selecting and working with large datasets, turning images into workable data for machine learning, and training and comparing convolutional neural networks with different learning rates. Initially, it seemed like a daunting task to determine which datasets were useful and which were not. Before deciding upon our final dataset, we tried using several other mask image datasets. We eventually realized that other datasets were lacking scale and diversity in images. Furthermore, we gained experience making design decisions such as selecting the loss function or post processing function for a machine learning model. An interesting aspect to our project was the comparison between the pretrained resnet model and our custom model which used 2D convolutional layers and max pooling. Our custom model was designed according to standards for machine learning with images as data, but the resnet model had the advantage of being trained on facial data beforehand and having significantly more layers. If we were to use more layers in our custom model, we expect that the results would be closer to the accuracy of the resnet model. In comparison to other projects that we had worked on in the past, this project was more open-ended and involved more creativity on our part. The process of going from an idea, to finding a dataset, to developing a working neural network was challenging yet rewarding. Nobody in our group had worked with pytorch or machine learning outside of this class and a single lab in computer architecture, so it was fulfilling to produce two working models for a 3-classification image problem in the end. In addition to the technical skills we developed, our group practiced technical writing and data formatting through the milestones.

Significant challenges

One of the most significant challenges throughout the project proved to be the time it took to train the models, especially resnet34. As a result, it was difficult to quickly run tests with varying parameters (ie. learning rate) to try to build the optimal network. To solve this challenge, we tested certain parameters and networks with a smaller dataset, which was a sample of the main dataset. This dataset set, stored in abridged_data.csv, allowed us to quickly run tests and extrapolate the results to test the network on a larger scale. With four members in our group, we tested the custom and resnet34 models by varying the learning rate by increments of .002. With each group member running two tests, we were able to efficiently test all the models and select the best-performing network. Although we would have liked to test a wider range of learning rates and a greater number of epochs, these tests still provided a good spread of data (see Appendix), which justified our decision to choose a learning rate of 0.007.

An additional challenge we faced was the lack of balance in the dataset. The data included a very large percentage of “face_with_mask” and “face_no_mask”. Obviously, this is the most relevant categorization as this would ultimately be the deciding factor for an actual system in the real world. Because detection of improper masking wearing would also be practical, we included this in our network; however, the sample of “face_with_mask_incorrect” wearing images was fairly small (150 images) compared to “face_with_mask” (4180 images) and “face_no_mask” (1569 images). As a result, it was more difficult to accurately predict the improper mask wearing. This is why we implemented an effective score to evaluate the network; because “face_with_mask_incorrect” and “face_no_mask” are practically similar and the network was only trained on 150 “face_with_mask_incorrect” images, we wanted to ensure that the network was given credit for an effectively correct guess. Initially, we also were thinking about training the network to recognize specific categories of masks such as surgical masks, neck gaiters, and about 10+ other mask types. These categories, similar to the improper mask wearing images, had very few images in the dataset compared. With such small samples, it would have been difficult to categorize so many types of masks within a short training time. Given a larger sample size of data, we could have trained the network to recognize all the mask categories.

If we were to expand the scope of the project and continue working on it in the future. DLib seems to be a promising addition to the custom neural network. DLib creates a map of points around key features of the face such as the nose and mouth to landmark features. If we could landmark key features that indicate the presence of lack of presence of a mask such as the nose and mouth, we could use this data to enhance our predictions. Specifically, in the improperly worn mask category we could apply this model. If a mouth was covered but the nose was exposed, this would be a strong indication that the mask is worn but worn improperly. DLib also works to recognize the face within an image that includes a landscape. Since our dataset had the face images labeled, this was unnecessary for the primary training and testing portion of the project; however, this could prove useful in the standardization of user submitted images in the demonstration phase of the project. (Martinez)

Compelling learning experience

This experience gave our group experience with artificial intelligence and gave us the confidence to undertake projects that involve AI. Through learning to use pytorch, we gained the knowledge necessary to train neural networks and use them to solve a problem. We also learned how to track the accuracy of a model through a variety of metrics. For our final deliverable, we used loss from the cross-entropy function, correct predictions, and our custom “effective correct” metric to evaluate how well the model is performing. Not only were we able to learn pytorch and the machine learning packages in python, but we were also able to apply the class concepts about neural networks effectively. Our best model was able to correctly make predictions in the three-classification problem around 96% of the time. It was meaningful for our group to apply artificial intelligence to a relevant real world problem and dataset. When approaching technical problems in the future, especially those containing image data, we will now have a baseline understanding of the requirements to design a convolutional neural network.

Works Cited

Alexander, Larry and Michael Moore, "Deontological Ethics", *The Stanford Encyclopedia of Philosophy* (Winter 2020 Edition), Edward N. Zalta (ed.), https://plato.stanford.edu/archives/win2020/entries/ethics-deontological/.

“An On-Device Deep Neural Network for Face Detection.” *Apple Machine Learning Research*, Web. 06 Oct. 2020. https://machinelearning.apple.com/research/face-detection.

Brownlee, J. (2019, July 05). A Gentle Introduction to Pooling Layers for Convolutional Neural Networks. Retrieved November 09, 2020, from https://machinelearningmastery.com/pooling-layers-for-convolutional-neural-networks/

Driver, Julia, "The History of Utilitarianism", *The Stanford Encyclopedia of Philosophy* (Winter 2014 Edition), Edward N. Zalta (ed.), https://plato.stanford.edu/archives/win2014/entries/utilitarianism-history/.

Face Mask Detection Dataset. (2020, June 14). Retrieved November 09, 2020, from https://www.kaggle.com/wobotintelligence/face-mask-detection-dataset

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." 25 Dec. 2015. Web. 16 Oct. 2020.

“How to Select, Wear, and Clean Your Mask.” (n.d.). Retrieved November 09, 2020, from https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/about-face-coverings.html

Jamil, N., et al. “Face Recognition Using Neural Networks.” *Proceedings. IEEE International Multi Topic Conference, 2001. IEEE INMIC 2001. Technology for the 21st Century*, IEEE, 2001, pp. 277–81, doi:10.1109/INMIC.2001.995351.

Kansal, Sachin. “Protection One Another.” *Uber Newsroom,* https://www.uber.com/newsroom/protecting-one-another/

Martinez, J. (2020, July 04). Detecting Face Features with Python. Retrieved November 11, 2020, https://towardsdatascience.com/detecting-face-features-with-python-30385aee4a8e

“No Face Mask Detection " Avigilon.” *Avigilon*, 14 Aug. 2020, www.avigilon.com/solutions/covid-19-resources/no-face-mask-detection

Norlock, Kathryn, "Feminist Ethics", *The Stanford Encyclopedia of Philosophy* (Summer 2019 Edition), Edward N. Zalta (ed.), https://plato.stanford.edu/archives/sum2019/entries/feminism-ethics/.

Pytorch.org, "CrossEntropyLoss." Web. 16 Oct. 2020.

Tryolabs. “Face Mask Detection in Street Camera Video Streams Using AI: behind the Curtain.” Tryolabs Blog, Tryolabs, 9 July 2020, tryolabs.com/blog/2020/07/09/face-mask-detection-in-street-camera-video-streams-using-ai-behind-the-curtain/.

Wood, T. (2019, May 17). Softmax Function. Retrieved November 10, 2020, from https://deepai.org/machine-learning-glossary-and-terms/softmax-layer

Appendix A: Mask and No Mask Classification Data

Average Testing Loss		
Learning Rate	Custom (Non Pretrained)	Resnet34
0.003	**0.004810**	0.008110
0.005	0.022742	0.00822
0.007	0.075870	**0.004435**
0.009	0.027353	**0.004481**
Average	0.03269375	0.0063115

Table 1.1 Average Testing Loss

Correct Predictions (Out of 1438 Possible)		
Learning Rate	Custom (Non Pretrained)	Resnet34
0.003	**1412 (98.19%)**	1396 (97.08%)
0.005	1296 (90.13%)	1398 (97.22%)
0.007	1092 (75.94%)	**1414 (98.33%)**
0.009	1286 (89.43%)	**1414 (98.33%)**
Average	1271.5 (88.42%)	1405.5 (97.74%)

Table 1.2 Correct Predictions

Appendix B: Mask, No Mask, and Mask Worn Incorrectly Classification Data

Average Testing Loss		
Learning Rate	Custom (Non Pretrained)	Resnet34
0.003	0.0383531	0.0148422
0.005	0.0461195	**0.0132650**
0.007	**0.0380162**	0.0135603
0.009	0.0467762	0.0207708
Average	0.04231625	0.015609575

Table 2.1 Average Testing Loss

Correct Predictions (Out of 1475 Possible)		
Learning Rate	Custom (Non Pretrained)	Resnet34
0.003	**1297 (87.93%)**	1420 (96.27%)
0.005	1253 (84.94%)	1423 (96.47%)
0.007	1270 (86.10%)	**1428 (96.81%)**
0.009	1245 (84.41%)	1409 (95.52%)
Average	1266.25 (85.84%)	1420 (96.26%)

Table 2.2 Correct Predictions

Effective Correct Predictions (Out of 1475 Possible)		
Learning Rate	Custom (Non Pretrained)	Resnet34
0.003	**1329 (90.10%)**	1441 (97.69%)
0.005	1292 (87.59%)	1442 (97.76%)
0.007	1306 (88.54%)	**1449 (98.24%)**
0.009	1278 (86.64%)	1437 (97.42%)
Average	1301.25 (88.22%)	1442.25 (97.78%)

Table 2.3 Effective Correct Predictions