

# Human-like Visual Perception of Realtime AI Agents

Evangelos Christopoulos

**Abstract**—Stealth mechanics in video games rely heavily on the ability of the NPC to visually identify the player in a realistic manner. Traditionally, this was achieved by a combination of several parameters, including the light level of the player’s surroundings and their stance. As effective as these parameters are in providing a predictable and reliable result, they miss out on more complex concepts like camouflage blending, silhouetting and occlusion which are key components of human pattern identification. The idea behind this paper is to train a Neural Network to take a low-resolution snapshot of the player’s character from the NPC’s point of view, and evaluate the confidence based on that snapshot. Combined with ground-truth values like the distance to the player and their movement speed, this approach should provide a more accurate confidence estimation than traditional methods, contributing to a more complex but immersive experience.

## I. INTRODUCTION

THE purpose of this paper is to compare different Neural Network architectures in their ability to provide realistic and readable confidence estimations of how well the NPC can distinguish the player among their surroundings. Unreal Engine 5 is used throughout this research.

## II. CAPTURE PROCESS

A Scene Capture Component was attached to the AI Character, configured to always point to the player, and the field of view is calculated to properly frame the player according to the following formula.:  
 $\text{Arctan}(y / \text{Distance}) * x$

Where Distance is the distance between the player and the NPC, y is the player character’s height and x functions as a simplified zoom multiplier. This ensures images are consistently centered around the player and convenient for use with neural networks, providing a stable baseline for consistent estimations. After the neural network provides its confidence estimation, it will be paired with distance and movement speed to provide the final confidence number.



**Fig. 1.** Captured image example

The Scene Capture outputs to a 32x32 Render Target. For performance reasons, this process is selectively taking place only when requested by the NPC’s logic, so for example when first establishing line-of-sight to the player or in set intervals while the player is within the NPC’s field of view.

## II. DATASET

In order to efficiently evaluate the performance of the neural network, common metrics would not suffice, as the model should provide human-like predictions instead of raw high accuracy. For this reason, it was decided to diversify the dataset based on level environments, player camouflage and armed state (whether the character is holding a weapon).



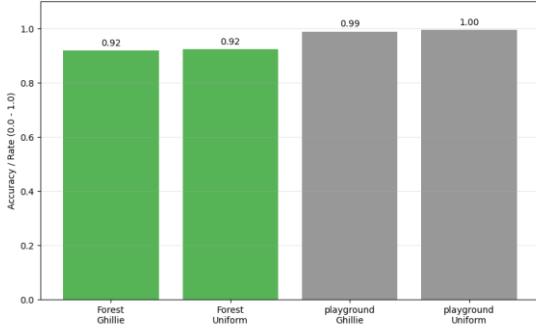
**Fig. 2.** Non-camouflaged, unarmed character (left), camouflaged, armed character (right)



**Fig. 3.** Forest level (left), playground level (right)

The reasoning is that a more human like model should display significantly higher confidence in the clean, playground environment compared to the complex forest. Similarly, the character with the uniform should produce higher confidence numbers in the forest compared to the camouflaged character. Lastly, although expected to have minor impact, the “unarmed” state is expected to yield slightly lower confidence.

During early testing however it was discovered that the playground environment yielded far higher confidence numbers with almost all architectures, as was expected. For this reason, it was decided to only capture the final dataset in the forest environment to simplify evaluation, as will be discussed below.



**Fig. 3.** Early results displaying far higher confidence in the playground environment

For more intuitive results, it was decided to capture a dataset of corresponding sets of 5 images per set taken under the same circumstances, but with the following differences:

- Camouflaged, Unarmed
- Camouflaged, not Unarmed
- Not Camouflaged, Unarmed
- Not Camouflaged, not Unarmed
- Player Hidden: this acts as the ‘0’ class

Each subset includes 750 indexed images, for a total of 3750 images. The index of each image defines the setting, meaning the predictions can be evaluated against the corresponding images from the other subsets.



**Fig. 4.** A subset of images and the predictions. Note how the images without the camouflage (3 and 4 from the left) provide significantly higher predictions than the ones with the camouflage. The model appears to rely on texture and color contrast similar to human perception, as is desired.

The dataset was captured by assigning an AI controller with a random walking task to the “player” character, then at set intervals running an Environment Query to find the farthest location with line-of-sight to the player. Then, an NPC assigned as a “capture” pawn teleports to that location, and performs the capture which is then exported to disk.

In order to create the subsets, due to the overhead introduced by the capture process, the game time had to be

stopped with TimeDilation (0) to allow for the different scenarios to be captured at different frames. This may introduce small artefacts and noise which must be taken into account during the evaluation process.



**Fig. 3.** Example of an artefact with one of the images

TimeDilation (100) is used when not capturing to allow the player to move long distances in short amounts of time without having to modify the character’s speed and thus the physics.

The scene is lit by a Directional Light which is assigned new height and rotation values every tick, simulating diverse lighting conditions.

The resulting Dataset is comprised of 3750 images across all 5 scenarios. This means a 4:1 imbalance between positive and negative labels, which will be taken into consideration in the evaluation process.

The images are loaded and parsed into “img” objects, which includes:

- img: image data (as read with opencv imread)
- X: extracted feature vector
- y: label
- prediction: for validation results assessment
- index: subset ID
- camo: camouflage metadata Boolean
- unarmed: unarmed metadata Boolean

The loading function reconstructs the label, index metadata from the folder structure.

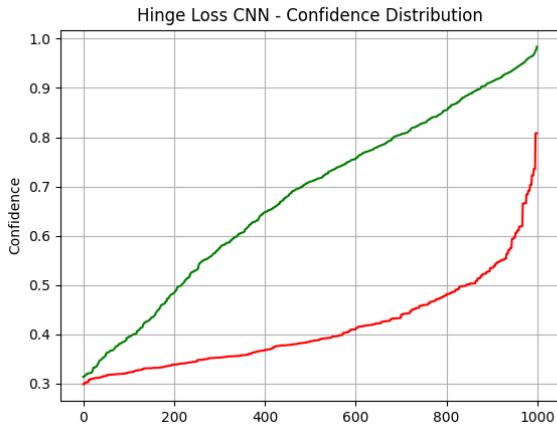
It was decided to train the models using the first 500 subsets, and evaluate them on the remaining 250.

### III. MODEL EVALUATION

With the assistance of the subsets developed, the performance of the models will be assessed based on the three following criteria:

#### 1) Confidence Distribution

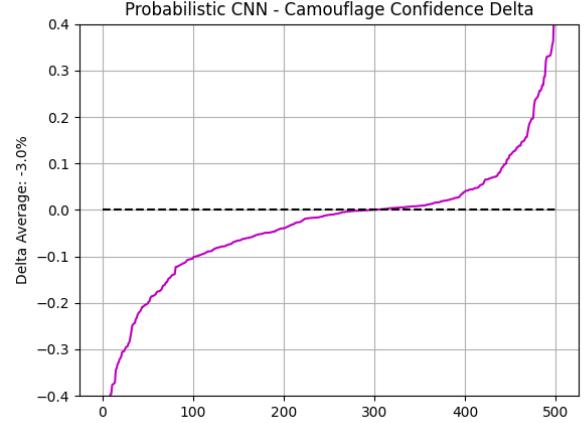
When plotting the predictions of the model per sample in increasing order, the model should provide good separation between the predictions of the images that contain the player (positive labels) and the images without the player (zero labels). It should be noted that “false positives” are not a concern to the final implementation of the model, since it is only meant to evaluate images that do contain the player, but they may function as an indicator that the model doesn’t hallucinate the player in the environment. Also, it is important for evaluation readability that the positive label predictions fall roughly into a linear pattern, since this way the difference between two predictions is more intuitively understood.



**Fig. 5.** Promising confidence distribution graph. The positive label confidence (green) is linearly distributed, and the negative label confidence (red) is significantly lower, indicating that the model doesn’t hallucinate.

#### 2) Camouflage Delta

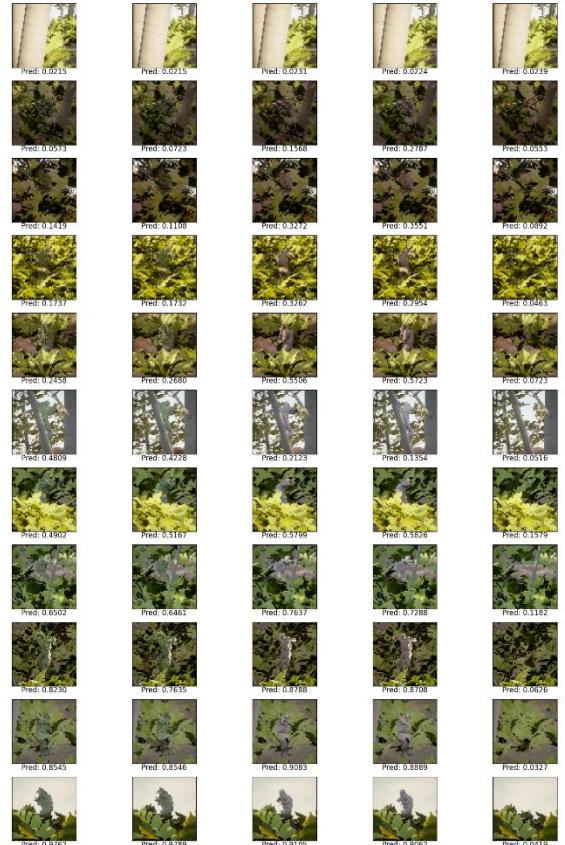
For each image subset, predictions of the camouflaged images are subtracted from the predictions of the non-camouflaged images. By plotting these differences in increasing order and calculating the average, the effectiveness of camouflage on the performance of the model can be visualized



**Fig. 6.** Camouflage Delta distribution graph. The -3% average value is a good indicator that the model displays lower certainty when dealing with a camouflaged player

#### 3) Visual Evaluation

Since the purpose of the models is to simulate human perception, there is no better measurement than visually inspecting the different probabilities assigned to different samples. Large example plots are necessary and inevitable for this type of evaluation.



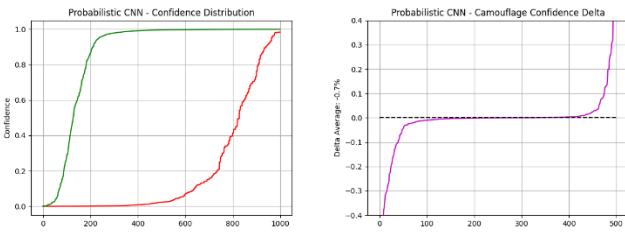
**Fig. 7.** Examples across varying prediction values

#### IV. NEURAL NETWORK ASSESSMENT

##### A. Probabilistic CNN

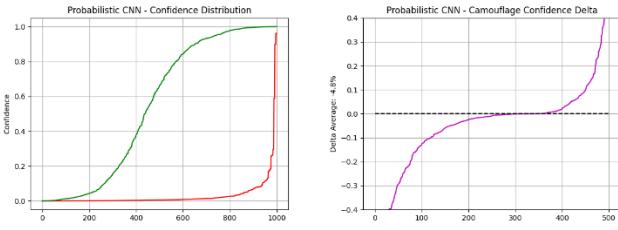
The first option assessed was a standard convolutional network with sigmoid activation. It was observed that the size of the final Dense layer played a major role in the performance of the model, reducing the Camouflage Delta Average from -1% at 512 filters to -10% at 128 filters on the architecture tested at the time.

It was also noted that the balanced class weights 4:1 tended to give high confidence numbers for both the positive and zero labels. The positive labels had an ~86% confidence average and the zero labels a ~20%. This meant unusable, high confidence distribution for the positive labels, but also led to a model that appeared invariant to camouflage.



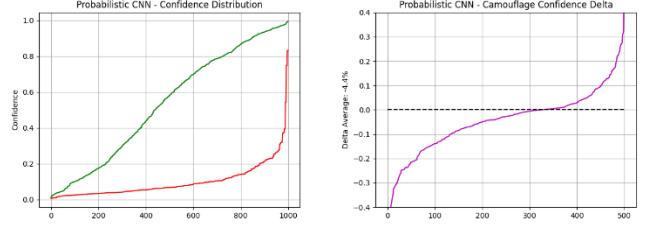
**Fig. 8.** Confidence Distribution and Camouflage deltas for the balanced model

By increasing the weight of the zero label (128 was found to be optimal), the confidence average of the positive labels fell to ~50%, and to ~3% for the zero labels. This created a smoother confidence distribution with fewer hallucinations, and interestingly increased drastically the effect of camouflage on the confidence.



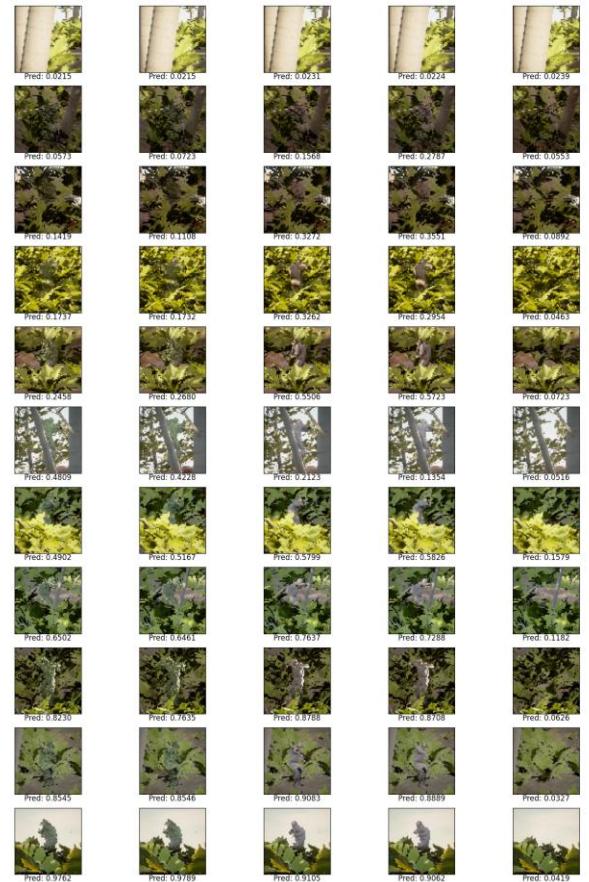
**Fig. 9.** Confidence Distribution and Camouflage deltas for higher zero label weights

However, the confidence distribution has a very distinct s-shape, leading to less diversified values close to the ends of the range. For this reason, temperature scaling was implemented, helping to smooth the curve while not significantly lowering the impact of camouflage.



**Fig. 10.** Confidence Distribution and Camouflage deltas with temperature scaling

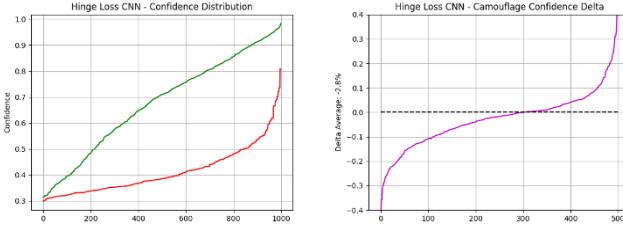
It is important to note that data augmentations were tested but were not implemented, as they tended to eliminate the advantage of camouflage and gave worse visual results.



**Fig. 11.** Visual inspection of the finished model. Note how the subsets include less occluded images where the character blends worse with the environment as the probability increases. Also, the camouflaged images generally show lower confidence, and the confidence of the empty images is almost zero. On the negative side, very similar images within subsets show significant fluctuations in confidence, meaning that although the model performs well, it is not particularly stable

## B. CNN with Hinge Loss

The second approach examined was to train the same CNN model with hinge loss instead of binary cross-entropy. It was observed that this method did not benefit from increased zero label weights, and although negative initial bias was used for the final Dense layer, it was not possible to get the same amount of separation between the positive and the zero labels.



**Fig. 12.** Confidence Distribution and Camouflage deltas for the final hinge loss model with temperature scaling applied

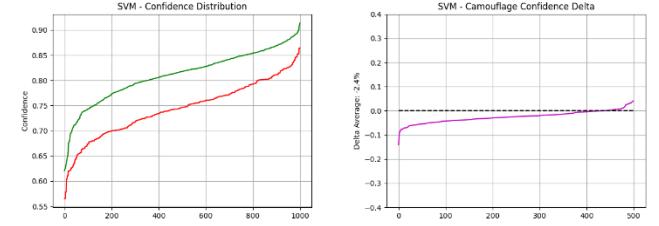
However, the camouflage impact curve is smoother than that of the regular CNN, indicating slightly less noisy predictions, as is confirmed by the subset examples.



**Fig. 13.** Examples of the hinge loss model’s performance. There is less variation between similar images, and more predictable variation within each subset. However, the high confidence of the zero labels raises questions as to the reliability of the model.

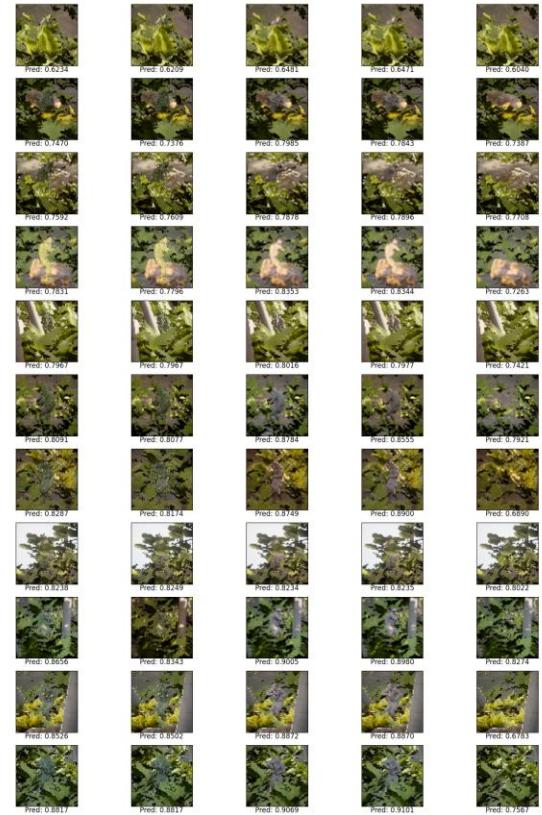
## C. Standard SVM Model

Although not expected to perform similarly to the first two approaches due to no spatial knowledge, a regular SVM model was tested on the raw image data, and the predictions were extracted using built in functions of the Scikit SVM library. Interestingly, this model was actually more resilient to noise, and more steadily influenced by camouflage.



**Fig. 14.** Although the confidence distribution is gathered between 60% and 90%, the camouflage affects detection around 2.5%. Also, the intersection point of the camouflage delta curve is moved to the right, as an indication that camouflage is now effective in more cases.

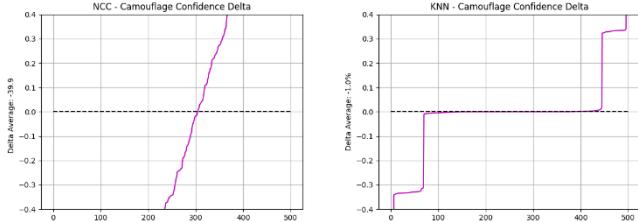
It was found that C=1 yielded better results than values between 0-1, while the RBF kernel was affected by camouflage when the Linear kernel was not.



**Fig. 15.** Examples of the simple SVM model. Predictions are much more stable between subsets thanks to the similarity in the data, however the predictions between different subsets are not reliable.

#### D. NCC and KNN comparisons

As anticipated, the NCC and KNN algorithms performed far worse than the previous approaches, with or without PCA, due to the noisy nature of the forest environment. Notably, camouflage impacted the performance of NCC while not the performance of KNN.



**Fig. 16.** Camouflage Deltas for NCC and KNN. Note that NCC results are not normalized, resulting in a 20-130 confidence range. Camouflage efficiency is still evident by the offset of the curve to the right.

#### V. SUMMARY

CNN models appear to be up to the task of achieving human-like, predictable confidence numbers that may offer the chance to replace regular algorithmic calculations with more meaningful and immersive models. It will be interesting to test these algorithms with larger, more diverse datasets across real game environments, to determine if noise in probabilistic CNN models drops or if the consistency of Hinge Loss models appears to be more vital.