Detection through smart cameras

In our project, we're trying to imitate the detection through smart cameras on different visions like detection of fire and smoke or combined with data from other sensors of the camera we can discover whether our app is still under our control or it's attacked by the others.

Here is an example camera architecture of Nest:

- Security: 128-bit AES with TLS/SSL 2-factor verification available
- Nest Cam IQ Indoor — indoor Nest camera with additional processing power, enabling sophisticated recognition, speaker, and snapshot features
- Where_id as unique of Nest generated identifier
- Recording last event url
- Providing recording of the camera instead of just alerts
- Use the wwn_security_state field to view the security state.

The above introduction is from Nest's website and it shows that the smart camera provides not only the recording of the vision of the camera, but also provides microphone to contact with others and sophisticated recognition to check whether stranger appears. It uses wwn_security_state to store the value of the state with OK and deter (person detected while away state). And the deter state is re-evaluated after several minutes to ok if no person exist and if the user is at home the state will be considered as OK.

From the background information of the smart cameras, it shows that it can connect to other sensors to check the state value to adjust the performance of the camera. For example, when the user is at home, it may turn off the recording feature to protect the user's privacy and it gets the state of user Home from other sensors to turn off its features.

In the project, our task is split into three parts as the detection of fire in users' house, checking the smart home is still under control by the users and the system and further detection of other disasters at home.

- Part 1: Fire Detection

In the first part, we focus on the detection of fire in or out of users' house. Due to the reason that in daily life users have the requirement of using fire like cooking or using stove, it needs to check whether it is a fire disaster or an accepted fire. As a result, we want to build models to let the camera get understanding of under what circumstance, it is a fire disaster need to be checked.

Figure a: An example of the comparison between input and output of the FireNet

a) Initial Training/Test Data Format:

In our project, it needs to imitate the performance of the smart cameras so that the training data should be series of images to be the examples transferred from the video. The training and test data are divided into five categories which are normal stove, normal cook, cook disaster, on fire and not on fire. These different inputs hope the output from the model to show it can detect fire in normal circumstances and ignore accepted fire during cooking and stove but not ignore a fire disaster in kitchen.

b) Data gathering

Based on the reason that a series of images may make the model more reliable, I try to catch videos from YouTube and then convert the videos into jpgs as a series of images. I gather some videos related to cook video, kitchen fire disaster and stove in home as the category of normal cook, cook disaster and stove. I also gather some outside fire videos to check the detection of fire without cooking and stove to protect the accuracy of predicting normal fire. Originally part of them images I'd like to use pix2pix to transfer them into night vision of images to check whether the model can also perform well at night. However, due to the implementation error of the pix2pix model, it doesn't work well for creating more input images. It will be delayed to later to try to succeed in transferring the images.

c) First Model Implementation

In our project, my method implements a two-step model prediction to guess whether the input image is on fire or not. For the first model, I implement FireNet model which is built based on YoLoV3. YoLoV3 is an object recognition model

which can recognize thousands of different objects, and FireNet is a subset of YoLoV3 which is a model focuses on fire detection. As a result, the first model will check whether the image is on fire or not. If the input data cannot pass the check of fire recognition, it means that mostly it is not on fire. If the model detects a fire in the image, it will output a value as [fire_percentage, [position_x, position.y, width, height] which is saved to be the second model input. Figure a is an example of the output of the model, which we can see that it circles the area where it detects fire and the more dangerous fire may have a bigger area which may help us determine the boundaries between accepted fire and fire disaster.

d) Second Model Implementation

In the second model, I build a logistic regression to make the prediction whether there is a fire alert need to be reported. I'm putting the value from the output of the first model combined with some other values which are assumed to set in the apps of smart cameras. For this test, I add two variables called cook and stove which display whether there's a stove or cooking exists in the vision of the smart cameras. Then send these values together as a new train data to the second model to get the prediction value of fire alert.

e) Evaluation

In my evaluation, I also divide my work into five parts like the training dataset. I want to check for every category whether it shows a good accuracy of every test to find what may be the vulnerabilities of the model.

1. Test on normal on fire object

For the first test, I'm using the FireNet test data as a test to show whether my new model will reduce the accuracy of the fire detection and it shows near 96% accurate which means it won't miss recognize the real fire.

2. Test on normal cook

For the second test, I'm using a series of cooking process images as input and make a test that with 92% accuracy that it won't consider a normal cooking process as a disaster.

3. Test on cook disaster

In my third test, I'm using a fire disaster in the kitchen to be the test images and it returns a value with 84%. I think this value which may needs to do further developed due to it's hard to find the boundary between a fire disaster and a normal fire. In my project, the predicted label which is between fire disaster and accepted fire is done manually. There may be further method to determine the boundary

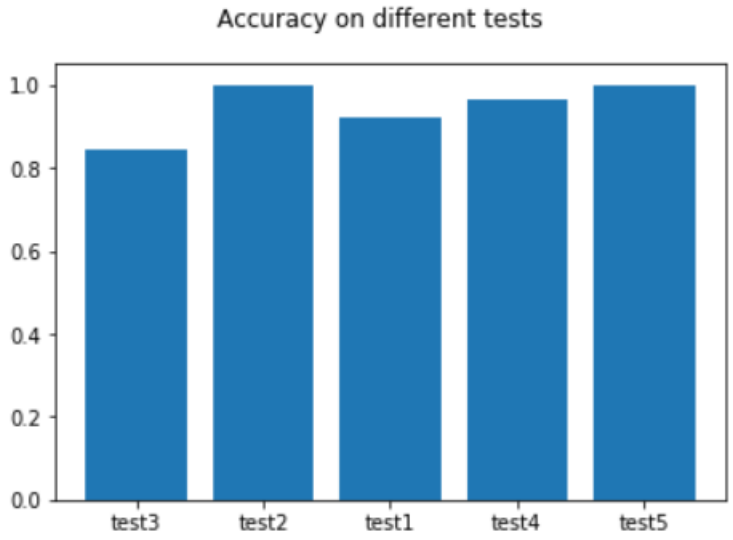
between a fire disaster and an accepted fire.

4. Test on stove

   In my fourth test, I'm using a series of stove picture to check whether it can detect stove fire. Due to the limitation of the training data, it shows that it can divides all stove fire, but it may have problem during fire disaster near a stove.

5. Test on No fire

   In my fifth test, I create some fake input with all value is zero as no fire pictures which shows that it won't detect no-fire.



The above image shows that the total accuracy of the five different tests. It's clear that despite the test on fire disaster detection, the remaining all has a high accuracy of detecting whether there is a fire alert. However, in test3, I think 84% is also reasonable to be implemented due to a series of images is done as a input so that it still has a high probability to detect the fire disaster.

f) Limitation and further development

1. Responding speed

   Due to the low efficiency of the YoLoV3, it needs to take few seconds to detect every images. As a result, it isn't acceptable to take every flash image to the model to do the detection. One possible solution is picking part of the series of images as the input to the model every few seconds instead of the total images which will reduce the responding time and still have high accuracy to detect fire alert.

2. It still needs a clearer definition between fire disaster and normal cooking fire due to in current data the output label is determined manually.

3. There are more circumstances can be added to the input of second training model like home/away state of the users. For example, it is impossible to have any fire

when the user goes outside so at that it will always respond a fire alert despite the cook signal.

- 2. Checking control of the smart home

The second part is something may be added to further development of the detection of the smart cameras. In part 1, what I have done is only detect the fire in users' house, but it can still be many other different evens occur like some stranger break into your home during you are outside. For example, if the attacker gets control of part of the smart app and change the house's state from away to Home, then the lock of the door is unlocked automatically due to the policy of the smart home. Then the attacker can go into user's room. However, if the camera isn't affected by the attacker which means that the camera is still in away mode, and it will detect that there's a stranger who breaks into the user's room during away state. After that the camera can send the information the cloud to check whether the smart home is lost control and let the cloud to make urgent services to protect the smart home.

As a result, my second part project may want to check how the camera can check the circumstance whether there are other people break the control of the smart home. In my basic scenario, I'd like to assume that the smart camera doesn't share the same sensor value with the smart home and it can interact with the cloud service to report the difference between the real state and fake state. In the vision of the camera, it can detect some strange people through object recognition which is also provided by Nest so that it can determines that the person at that time is not the user but the attacker. Then after that time the camera will send the combination of the attacker's image and the current state to the cloud service to show that the smart home is lost control.

Furthermore, there are still different sides of attacks to smart homes, but I'd like to check the camera output is correct always to show that the camera can be an evidence to determine whether others has modified the monitored area.

- Part 3: Detection of other bad accident events at home

Despite the fire, there is also many different accidents can be detected by the camera such as water leak, forgetting to turn off windows during the typhoon etc. A Although it's hard to detect all kinds of unpredictable events around us, it will be better to try to detect as much as the camera can to protect the user's security.

For this part, I still don't have a clear idea how I can protect as many kinds of accidents as I can protect the smart home through the camera, so it may be delayed to future days development.

- Conclusion

In the project, I think it succeeds in most part of detect fire through smart camera, and it also performs well on distinguishing between accepted fire at home and fire disasters at home.

For further development, it may need to take more time to think about how I can detect different kinds of accidents due to it's impossible to continue using YoLoV3 to detect different objects in the vision which is low efficiency. If I want to continue using YoLoV3 as an transferring model, I think that it may be an option to detect all objects

in the vision of the camera when it’s at the free time, and it will be updated every few hours or asked by the users, then the camera can know where the objects are then it may be easier for the camera to detect it’s the user is washing or it may be a water leakage, and it can also get control of the position of the windows.

Reference paper

"Security : Nest Developers." Nest Developers, developers.nest.com/guides/api/security-guide.

"Camera : Nest Developers." Nest Developers, developers.nest.com/guides/api/camera-guide.

Tsang, Sik-Ho. "Review: YOLOv3 - You Only Look Once (Object Detection)." *Medium*, Towards Data Science, 20 Mar. 2019, towardsdatascience.com/review-yolov3-you-only-look-once-object-detection-eab75d7a1ba6.

OlafenwaMoses."OlafenwaMoses/FireNET." GitHub, 20 Aug. 2019, github.com/OlafenwaMoses/FireNET.

Z. Jiao *et al*., "A Deep Learning Based Forest Fire Detection Approach Using UAV and YOLOv3," *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*, Shenyang, China, 2019, pp. 1-5, doi: 10.1109/ICIAI.2019.8850815.

Li, Pu, and Wangda Zhao. "Image Fire Detection Algorithms Based on Convolutional Neural Networks." Case Studies in Thermal Engineering, Elsevier, 10 Mar. 2020, www.sciencedirect.com/science/article/pii/S2214157X2030085X.

Images:

https://www.youtube.com/watch?v=K7cELdaxGdA
https://www.youtube.com/watch?v=iQrmJqt8mlM
https://www.youtube.com/watch?v=AFwkGTEles8
https://www.youtube.com/watch?v=6E2d-bVKfNk
https://www.youtube.com/watch?v=2yiH49K7qfw
https://www.youtube.com/watch?v=gLnCRBgYOAk
https://github.com/OlafenwaMoses/FireNET