

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Adversarial Attacks on Aerial Scene Classification**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Shorya Sharma**

School of Electrical Sciences

Indian Institute of Technology Bhubaneswar

[ss118@iitbbs.ac.in](mailto:ss118@iitbbs.ac.in)

# Abstract

Remote sensing image (RSI) scene classification is the foundation and important technology of ground object detection, land use management and geographic analysis. During recent years, convolutional neural networks (CNNs) have achieved significant success and are widely applied in RSI scene classification. However, crafted images that serve as adversarial examples can potentially fool CNNs with high confidence and are hard for human eyes to interpret. For the increasing security and robust requirements of RSI scene classification, the adversarial example problem poses a serious problem for the classification results derived from systems using CNN models, which has not been fully recognized by previous research. In this study, to explore the properties of adversarial examples of RSI scene classification, we create different scenarios by testing 29 major attack algorithms trained on different RSI benchmark datasets (NWPU-RESIC45 and UC Merced Land Use Dataset) to fool CNNs (i.e., AlexNet, ResNet50, ResNet101, MobileNetV2 and DenseNet121). We also explored the transferability properties of Transferable Sparse Adversarial Attack (a bench-mark black box adversarial attack with high transferability rate under L0 conditions) in this study. Attack selectivity reveals potential classes of adversarial examples and provides insights into the design of defensive algorithms in future research.

# Introduction

With the advancement of remote sensing technology, the automatic interpretation of remote sensing images (RSIs) has greatly improved [1]–[4]. RSIs with higher resolution have led to the wide use of RSI scene classification systems in crop classification [5]–[7], forest resource surveys [8]–[10], land cover classification [11], [12], building detection [13], [14] and other fields [15], [16]. Automatic interpretations of RSIs with robust and high accuracy can deliver high economic efficiency [17]–[19].

A good image classification algorithm can effectively extract target features and quickly locate new targets, which is the key to RSI interpretation. Therefore, the variability in features can be massive among different classification tasks. most traditional RSI interpretation algorithms design the hand-craft features for different applications, such as road detection [20], [21], vegetation measurement [22], drought monitoring [23], etc. [24]–[26]. However, these well-designed features cannot solve the RSI interpretation problem well since traditional algorithms need to be carefully designed with solid domain knowledge in order to be effective in different applications [27]–[29]. And most traditional classification algorithms have a low classification accuracy and calculation inefficiency, which makes it difficult to meet the system requirements.

In recent years, deep learning has achieved remarkable progress in the field of computer vision [30]. Many excellent algorithms, especially convolutional neural networks (CNNs), have been introduced into the RSI scene classification systems [5], [11], [15]. Chen *et al.* [31] used a single-layer autoencoder (AE) and multi-layer stacked autoencoder (SAE) to learn the features of RSIs and proposed a classification method using the extracted spatial features. This method performs better than do support vector machines (SVM) and K-nearest neighbor (KNN) classification models. Chen *et al.* [32] proposed a deep belief network (DBN)-based image classification framework that combined the spectral and spatial features of RSIs and used a CNN model to extract robust features from images, thereby greatly reducing the number of hyperparameters. To alleviate the overfitting problem of CNNs, Cheng *et al.* [33] added the metric learning regularization term to the CNN model by optimizing the discriminant objective function. This approach efficiently reduced classification error. Zhang *et al.* [34] proposed a gradient boosting random convolution network that reused the weight of each CNN model and reduced the parameters, thus making feature extraction more efficient. Chaib *et al.* [35] used the CNN model in feature extraction and discriminant correlation analysis (DCA) for data fusion. These RSI scene classification models perform much better in terms of accuracy than traditional RSIs and in the efficiency of RSI scene classification achieved using high-performance computing. Meanwhile, previous research has also demonstrated the excellent feature extraction abilities of CNNs.

However, CNNs have many limitations and still have room for improvements, especially concerning security problems [36]–[39]. Szegedy *et al.* [40] find that a small adversarial perturbation can be achieved on a trained model with a gradient algorithm. This happens when we deliberately add noises to the input images to fool the CNN classifier and prompt it to make wrong predictions with high confidence. The modified images are called adversarial examples.

The problem of adversarial examples exists in various application areas of deep learning [41]. The classification problem of RSIs presents a great security risk, especially for applications in the military and automatic driving fields [42]. As shown in Figure [1](#_bookmark0), in military applications, an attacker can generate an adversarial perturbation for a target such as an airplane to camouflage the object using a physical object. The adversarial example of the target can easily escape the detection by the military target detection systems. In this case, the originally robust target would be wrongly classified as the wrong class with high confidence. State-of-the-art RSI scene classification systems depend heavily on CNNs to extract features from the target. Unlike RGB images, RSIs have unique properties, such as spectra, bands, etc. In addition, RSIs are usually obtained from the nadir perspective, without the foreground and back views of natural images. Therefore, it is necessary to study the characteristics of adversarial examples in RSI scene classification systems.

In the experiment, we train several CNN models with the widest application in RSI scene classification systems. In these high-accuracy CNNs, we use a variety of attack algorithms to generate different adversarial examples. The result shows the vulnerability of several mainstream CNNs and demonstrates the importance of adversarial examples in RSI scene classification. Furthermore, we find that the fundamental issues related to adversarial examples of RSIs are the model vulnerability and attack selectivity. This means that different RSI scene classification CNNs have differ- ent security characteristics, so the cost to obtain adversarial examples varies. We also find that attack selectivity is related to the model structure and the training data, producing a clustered pattern of errors in generating adversarial examples. The misclassified adversarial examples are highly similar to the correct original scenes. These properties also reflect the characteristics of adversarial examples in RSI scene classification.

The main contributions of this article are listed as follows:

* We implemented white-box adversarial attacks using 5 different neural network architectures on different aerial scene classification datasets under two different configurations: eps=0.0005 and eps=1.0.
* We tabulated our results and inferred the best-attack on each dataset using a particular network architecture. Confusion Matrix were also plotted for evaluation purposes.
* We implemented the Transferable Sparse Adversarial Attack (TSAA) and deployed it on our dataset to infer the transferability of this black-box attack on our datasets using different neural network architectures.
* We implemented state-of-the-art black box attacks on our datasets using Adversarial Attack libraries.

# Related work

In recent years, CNNs have been applied in many tasks in the field of remote sensing, such as landmark classification [43], land use classification [44] and change detection [45]. CNNs have produced outstanding results in these areas. However, most of the research focuses on how to improve the model accuracy and design good CNN structures [46]–[48], and insufficient attention has been placed on the adversarial example problem of RSI scene classification systems. Since the adversarial example was proposed by Szegedy *et al.* [40], the security of deep learning has been the subject of widespread discussion.

The adversarial example refers to an input image formed by adding a special adversarial perturbation, which results in the model producing a false output with high confidence. Generally, a small adversarial perturbation is not perceived by the human eye. Goodfellow *et al.* [37] stated that adversarial examples in CNNs are produced due to the linear operations of the model in a high-dimensional space. For feature vector operations in a high-dimensional space, small perturbations can cause substantial errors by reducing the robustness of the model. This problem not only is encountered in computer vision tasks but also exists in many fields such as natural language processing and sentiment analysis [49]. It is well known that RSIs consist of high-dimensional data, so the problem of adversarial examples in RSI scene classification systems is also unavoidable.

The attack algorithms of adversarial examples are also diverse. According to the number of iterations, attack algorithms can be divided into the one-step attack and iterative methods. These attack algorithms are gradient-based optimization methods that try to maximize the loss of the objective function. The fast gradient notation (FGSM) method is a one-step attack proposed by Goodfellow *et al.* [37], which first computes the gradient direction of the model loss function and subsequently increases the value of the loss function by adding a small adversarial perturbation in that direction. Thus, the prediction result of the model deviates from the real class. This attack algorithm is simple, but the fooling rate of deceived models is low. Therefore, an iterative method named basic iterative method (BIM) is proposed [38]. The algorithm is an extension of the FGSM, and with multiple iterations, it attempts to add noise in each step to increase the value of the loss. In each iteration, the changed image pixels are controlled within a certain neighborhood of the original picture. Different from the FGSM, the BIM algorithm can effectively improve the fooling rate of attacks. FGSM and BIM are the two most widely applied attack algorithms.

Attack algorithms can also be divided into other types. Attack algorithms can be divided into white-box attacks and black-box attacks [36] according to the model information acquired by the attacker. In a white-box attack, the attacker knows the type of CNN, the number of layers, and the optimization method used for training and can obtain the training data and even discern the hyperparameters of the model. Attackers can launch additional targeted attacks based on the details of the model. In contrast to white-box attacks, attackers in black-box attacks are unaware of the model details. They can use only the model application background and outputs to analyze the vulnerability of the model. According to the characteristics of the attack algorithm, Prem-latha *et al.* [50] and Chakraborty *et al.* [51] classified attack scenarios into three categories: evasive attacks, poisoned attacks and exploratory attacks. Attackers in an evasive attack attempt to avoid detection by the system by constructing malicious input. In this scenario, the attacker cannot influence the training data of the model. The attacker in a poisoned attack scenario attempts to inject constructed malicious data to poi- son the model during the training process. In the exploratory attack scenario, the attacker cannot affect the training dataset or the details of the model and acquires knowledge of the learning algorithms and training data by testing the response of the model to the input data, thereby constructing effective adversarial examples. According to whether the attack class is clear, attack algorithms are also divided into targeted attacks and untargeted attacks [36]. Targeted attacks attempt to make the model classify the adversarial example as a specific class, whereas untargeted attacks do not care about the prediction labels of adversarial examples. Untargeted attacks require only the model to misclassify data that were originally correctly classified.

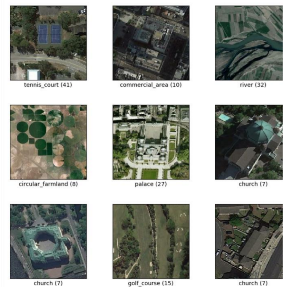
Above all, adversarial example attacks might occur during training and testing of a model. The type of attack is related to the model details and training data. The type of attack algorithm depends on the setting of the attack scenario. For example, FGSM and BIM can be both white-box and black-box attack algorithms. It depends on whether the information (e.g., model structure, hyperparameters) of the attacked model is known before the attack. Similarly, FGSM and BIM can also be targeted and untargeted attacks, depending on whether the image is misclassified into a specific class. Under different attack types, the object function of the attack algorithm changes. For example, the FGSM targeted attack algorithm only needs to change the loss function into the optimization of a specific class. In our research, we use all white-box, untargeted attack algorithms.

Currently, RSI scene classification involves many CNN-based applications that might also have adversarial example problems. These problems could have highly serious consequences for the application of RSI scene classification. Moreover, since RSIs have spatial and spectral properties, they are different from the RGB images. Therefore, the adversarial example problem is of great importance in RSI scene classification.

# Data

Two Datasets were used:

* RESISC45 dataset is a publicly available benchmark for Remote Sensing Image Scene Classification (RESISC), created by Northwestern Polytechnical University (NWPU). This dataset contains 31,500 images, covering 45 scene classes with 700 images in each class.
* UC Merced is a 21-class land use remote sensing image dataset, with 100 images per class. The dataset contains 2100 images which were manually extracted from large images from the USGS National Map Urban Area Imagery collection for various urban areas around the country. The pixel resolution of this public domain imagery is 0.3 m



NWPU-RESIC 45 UC Merced Dataset

# Method

This section gives a brief introduction of CNNs and the most widely used robust CNN model structure in RSI scene classification. Subsequently, it explains how to use attack algorithms to generate an adversarial example of an RSI.

1. *CONVOLUTIONAL NEURAL NETWORKS*

# Current RSI scene classification systems have better classification capabilities, and they are mostly based on CNNs. CNNs usually consist of convolutional layers, pooling layers, and fully connected layers [52]. Each layer has a different function. Through these structures, RSIs are continuously reduced in dimension, and semantic features are gradually abstracted. Eventually, the model implements the classification of the RSIs.

# Pooling operations can reduce the dimensions of the RSI to reduce the computational complexity and improve the generalization ability of the model. Finally, *Al* denotes the feature maps produced after the convolution and pooling operations. When the feature maps are fed into the fully connected layer, all of the feature maps need to be flattened and trans- formed into vectors. Finally, A classifier can predict a class *y* based on the final outputs. When the prediction is not correct, we use the cross-entropy loss function to compute the distance between the predicted results and the real labels. The model can be updated by the backpropagation algorithm [52], which aids the model in reducing the loss value. The model gradually converges in the process of iteration

# AlexNet

# In 2012, *Alex Krizhevesky* and others proposed a deeper and wider CNN model compared to LeNet and won the most difficult ImageNet challenge for visual object recognition called the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012 [7]. AlexNet achieved state-of-the-art recognition accuracy against all the traditional machine learning and computer vision approaches. It was a significant breakthrough in the field of machine learning and computer vision for visual recognition and classification tasks and is the point in history where interest in deep learning increased rapidly.

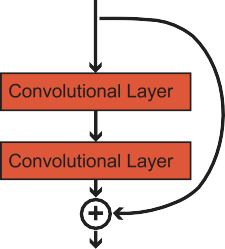
# The architecture of AlexNet is shown in Fig. 14. The first convolutional layer performs convolution and max pooling with Local Response Normalization (LRN) where 96 different receptive filters are used that are 11×11 in size. The max pooling operations are performed with 3×3 filters with a stride size of 2. The same operations are performed in the second layer with 5×5 filters. 3×3 filters are used in the third, fourth, and fifth convolutional layers with 384, 384, and 296 feature maps respectively. Two fully connected (FC) layers are used with dropout followed by a Softmax layer at the end. Two networks with similar structure and the same number of feature maps are trained in parallel for this model. Two new concepts, Local Response Normalization (LRN) and dropout, are introduced in this network. LRN can be applied in two different ways: first applying on single channel or feature maps, where an N×N patch is selected from same feature map and normalized based one the neighborhood values. Second, LRN can be applied across the channels or feature maps (neighborhood along the third dimension but a single pixel or location).

# AlexNet has 3 convolution layers and 2 fully connected layers. When processing the ImageNet dataset, the total number of parameters for AlexNet can be calculated as follows for the first layer: input samples are 224×224×3, filters (kernels or masks) or a receptive field that has a size 11, the stride is 4, and the output of the first convolution layer is 55×55×96. According to the equations in section 3.1.4, we can calculate that this first layer has 290400 (55×55×96) neurons and 364 (11 ×11×3 = 363 + 1 bias) weights. The parameters for the first convolution layer are 290400×364 = 105,705,600. Table II shows the number of parameters for each layer in millions. The total number of weights and MACs for the whole network are 61M and 724M respectively

# 

1. Resnet

Another model structure, ResNet [57], won the 2015 ILSVR competition, as shown in Figure 3(b). The model takes the depth to extreme values, and for the first time, the model achieved a depth of 152 layers. ResNet50 is the 50-layer version of ResNet. To train deeper structures, ResNet introduced a residual module, as shown in Figure 3(a), which implements the delivery of information between layers through a shortcut connection. Without adding parameters, this model can directly place shallow features into the upper layer, thus greatly increasing the training speed of the model. ResNet achieved a milestone in the ImageNet competition [56] and was the first model to surpass human classification in the ImageNet classification task. Moreover, as the main backbone of the detection model, ResNet has achieved good results for the VOC Pascal [58] and COCO [59] datasets and in other detection tasks [60]. Many RSI detection appli cations are based on ResNet [61].

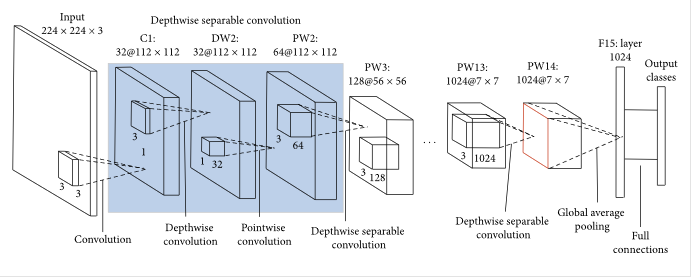




1. MobileNet

MobileNet model is designed to be used in mobile applications, and it is TensorFlow’s first mobile computer vision model. MobileNet uses **depthwise separable convolutions.**It significantly**reduces the number of parameters**when compared to the network with regular convolutions with the same depth in the nets. This results in lightweight deep neural networks.

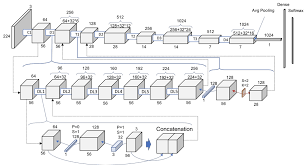
A depthwise separable convolution is made from two operations: Depthwise Convolution and Pointwise Convolution. MobileNet is a class of CNN that was open-sourced by Google, and therefore, this gives us an excellent starting point for training our classifiers that are insanely small and insanely fast.



1. DenseNet

In a normal feedforward network, an output from a layer is passed on to only the next occurring layer and as the network goes deep it would have access to only the higher-level features but not those from lover levels closer to input. In Densenet every subsequent layer receives input from all the previous layers in the chain as can be observed from above figure. All these values are concatenated so that information is not lost. The concatenated feature map is passed through a composite function consisting of Batch Normalization, Relu and 3x3 Convolution. The output is then passed on to every subsequent layer and the process is repeated.

But simple concatenation can lead to very huge number of parameters in a deep network. So the architecture is divided into multiple dense blocks each separated by a block called transition block. In dense block the information flows in the way described in above paragraph. Transition block is the place where downsampling occurs to prevent blowing up. The transition layer consists of batch normalization layers followed by 1x1 convolution layer which is followed by 2x2 average pooling layer.

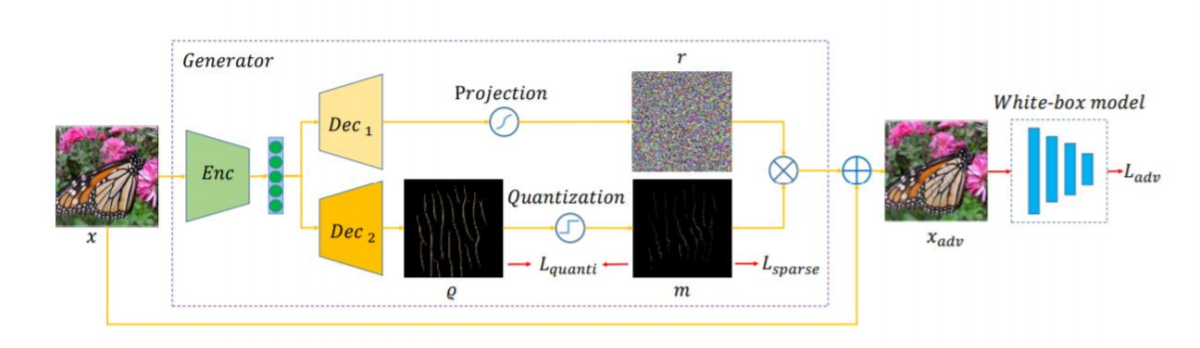


1. *ATTACK ALGORITHM*

These excellent models still cannot escape the problem of adversarial examples. The goal of the adversarial example is primarily to make the model misclassify the minimally changed image when the original image can be classified correctly. The attack algorithms of adversarial examples are also applicable to RSIs.

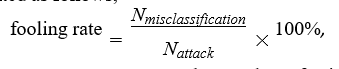
* 1. White-Box Adversarial Attacks
* Gradient-Based Attacks: Gradient Attack, Gradient Sign Attack (FGSM), Iterative Gradient Attack, Iterative Gradient Sign Attack, DeepFool L2 Attack, DeepFool L∞ Attack, L-BFGS Attack, SLSQP Attack, Jacobian-Based Saliency Map Attack
* Score-Based Attacks: Single Pixel Attack, Local Search Attack, Approximate L-BFGS Attack
* Decision-Based Attacks: Boundary Attack, Pointwise Attack, Additive Uniform Noise Attack, Additive Gaussian Noise Attack, Salt and Pepper Noise Attack, Contrast Reduction Attack, Gaussian Blur Attack, Precomputed Images Attack
  1. Transferable Sparse Adversarial Attack (Black-Box Adversarial Attack)

TSAA is a sparse adversarial attack based on the L0 norm constraint, which can succeed by only modifying a few pixels of an image. Prior sparse attack methods achieve a low transferability under the black-box protocol because they methods rely on the target model’s accurate gradient information or its approximation, causing the generated adversarial examples overfitting the target model. TSAA is a trainable generator-based architecture which tends to alleviate the overfitting issue by learning to translate a benign image into an adversarial example and thus efficiently craft transferable sparse adversarial examples



* 1. Fooling Rate

To evaluate the effects of attack algorithms, we introduce the definition of the fooling rate [36], which reflects the model’s ratio of misclassified images to attack images and can be represented as follows,

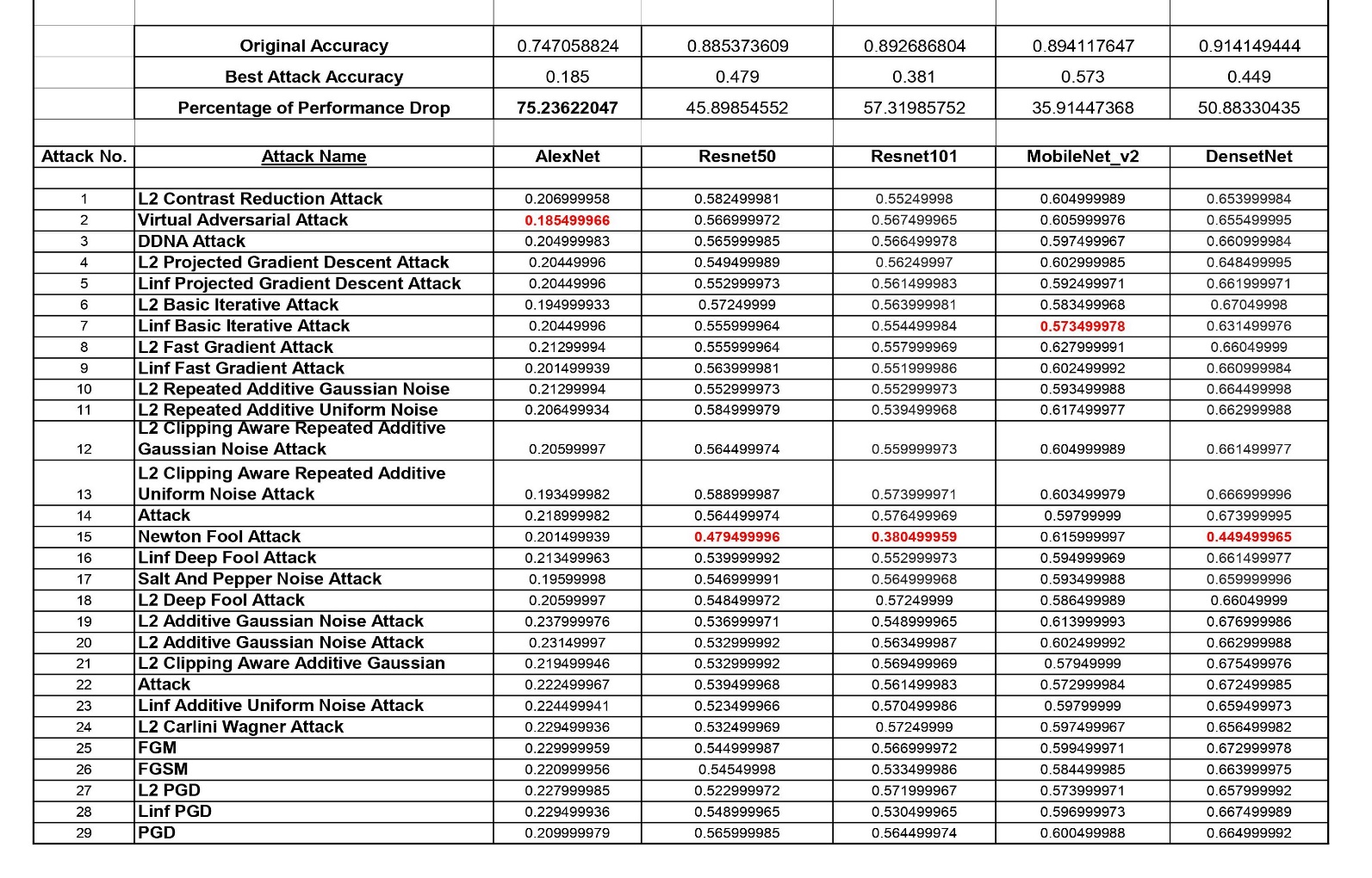


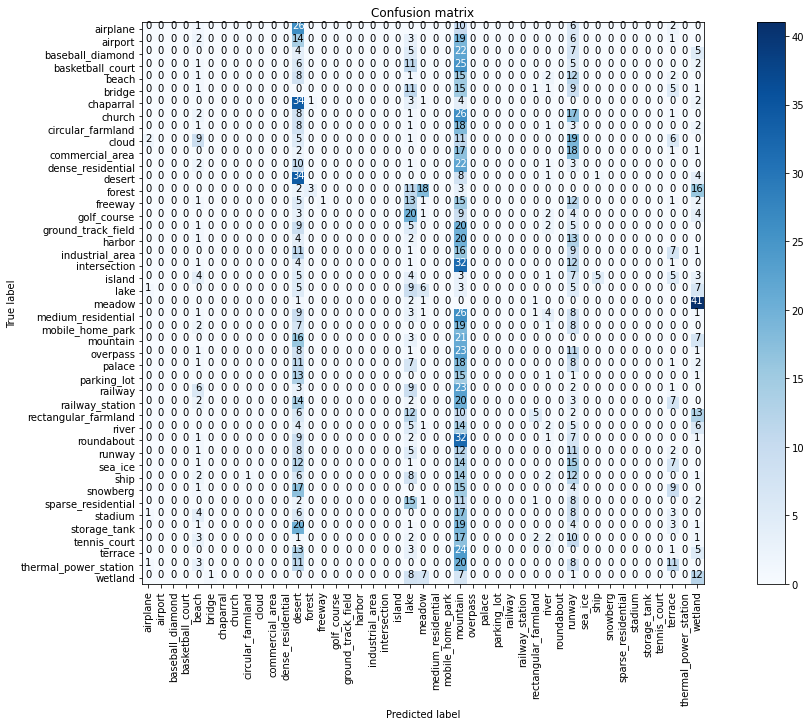
where *Nmisclassification* represents the number of misclassified images after the attack. Similarly, *Nattack* represents the number of images attacked. The fooling rate represents how many adversarial examples can successfully fool the model. It can be used to evaluate both models and attack algorithms. For the same model, a strong attack algorithm can obtain more adversarial examples that can fool the model with a higher fooling rate. For the same attack algorithm, a high fooling rate indicates that the model has poor robustness.

# Experiments

# In this section, we first use the white-box attack algorithms to attack multiple converged RSI scene classification models under different datasets. Second, we analyze the transferability of TSAA. The attack object of the adversarial example is a model with high accuracy that has been converged. In the experiment, we use three commonly used RSI datasets, including UC Merced Land Use [62], NWPU-RESISC45 [63]. For all datasets, we choose 80% as a training set and the remaining 20% as a test set during the training progress. For deep learning models, we choose AlexNet, ResNet50, Resnet101, MobileNetV2 and DenseNet121 which are widely used in RSI scene classification systems. The experimental platform is based on Ubuntu 16.04, with a 12 × 3*.*20 GHz Core i5 CPU, Tesla M60 GPU and PyTorch Framework. Cross-Entropy Function was used as the loss function with Adam Optimizer being selected for optimizing the weights. Confusion Matrix was used as the evaluation Metrics along with accuracy.

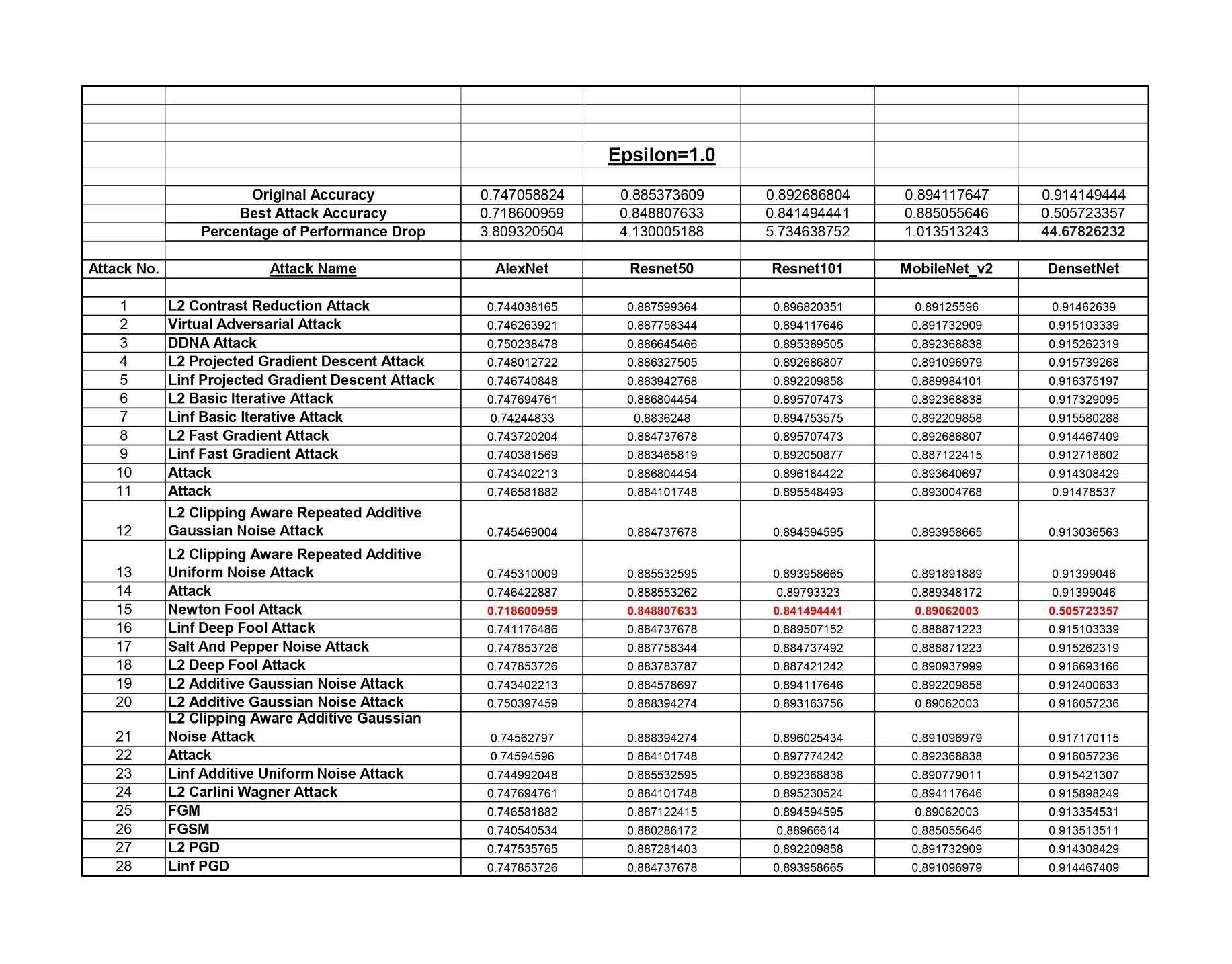
1. *White-Box Attacks*
2. NWPU-RESIC45

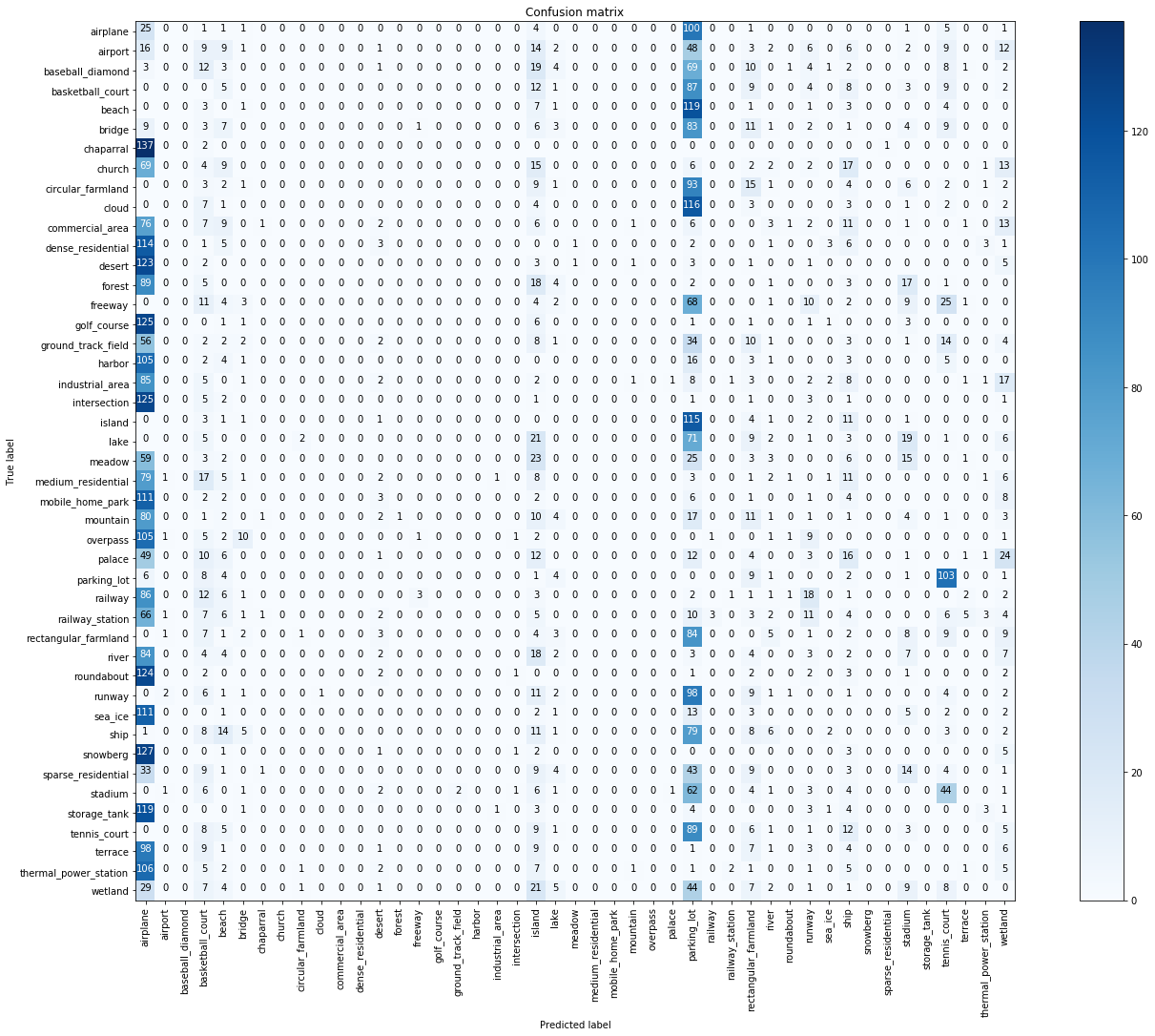
* Epsilon: 0.0005



Confusion Matrix for Virtual Adversarial Attack on AlexNet

* Epsilon: 1.0

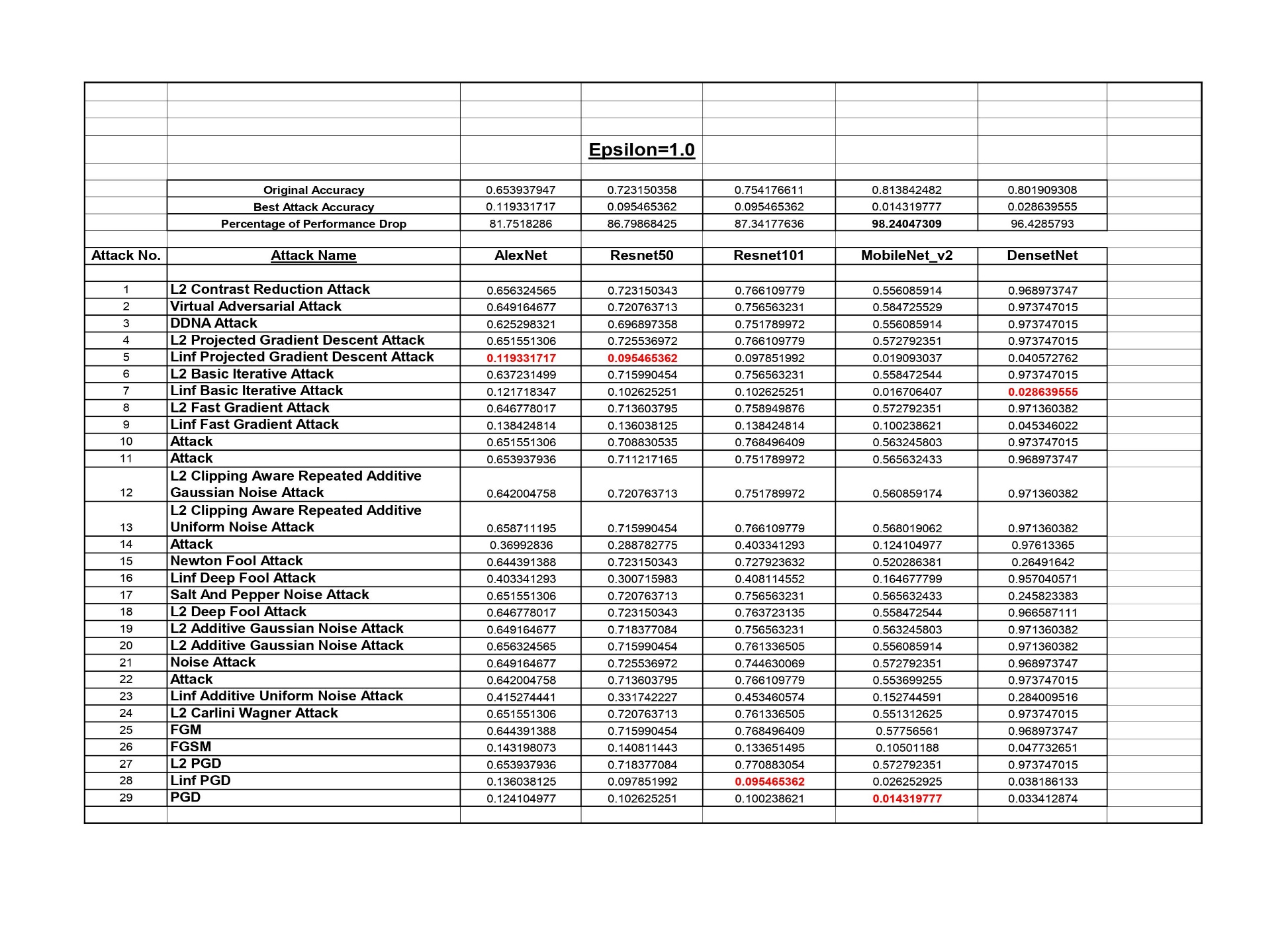


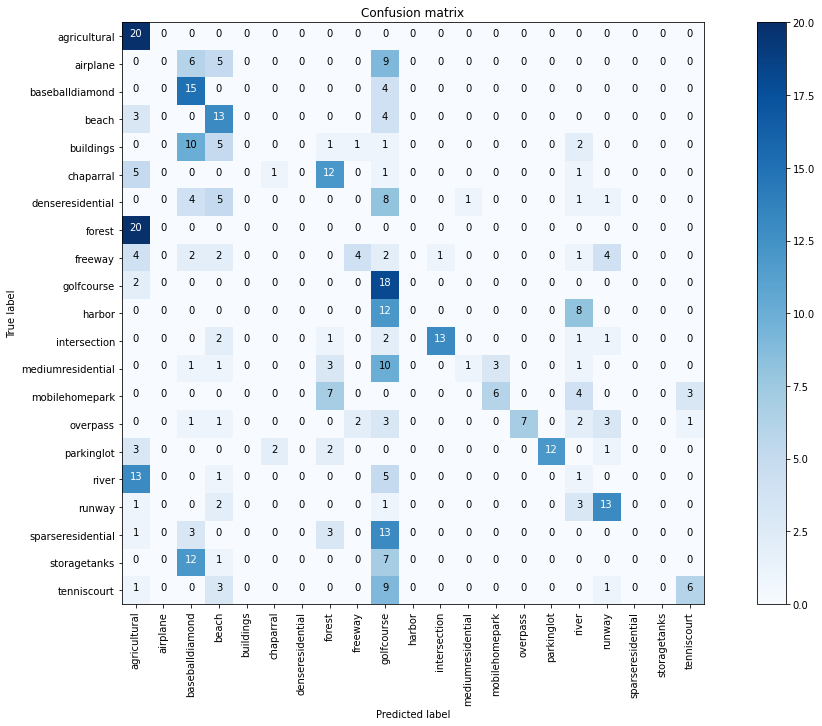


Confusion Matrix for Newton Fool Attack on DenseNet121

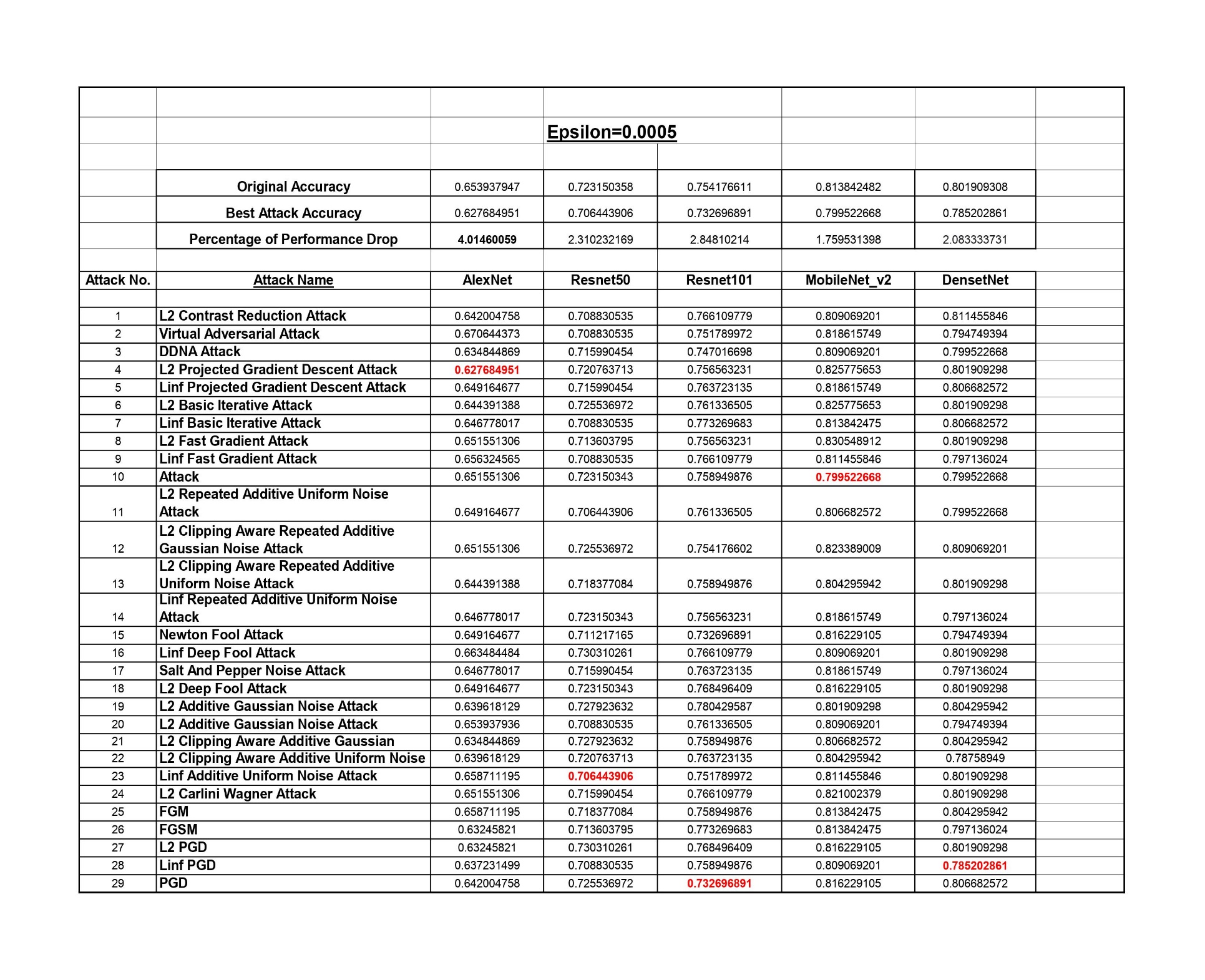
1. UC Merced Land Use Dataset

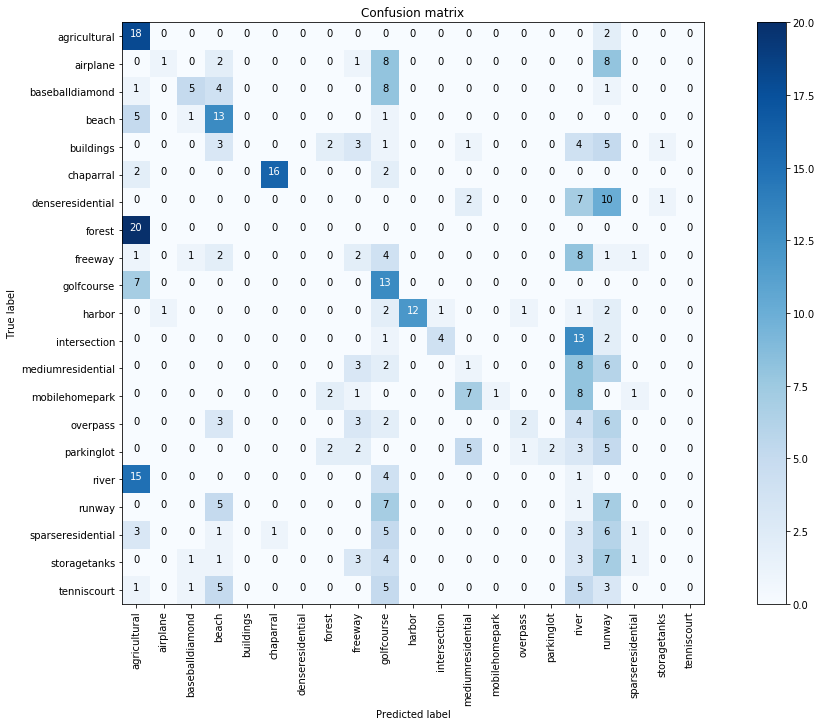
* Epsilon: 0.0005





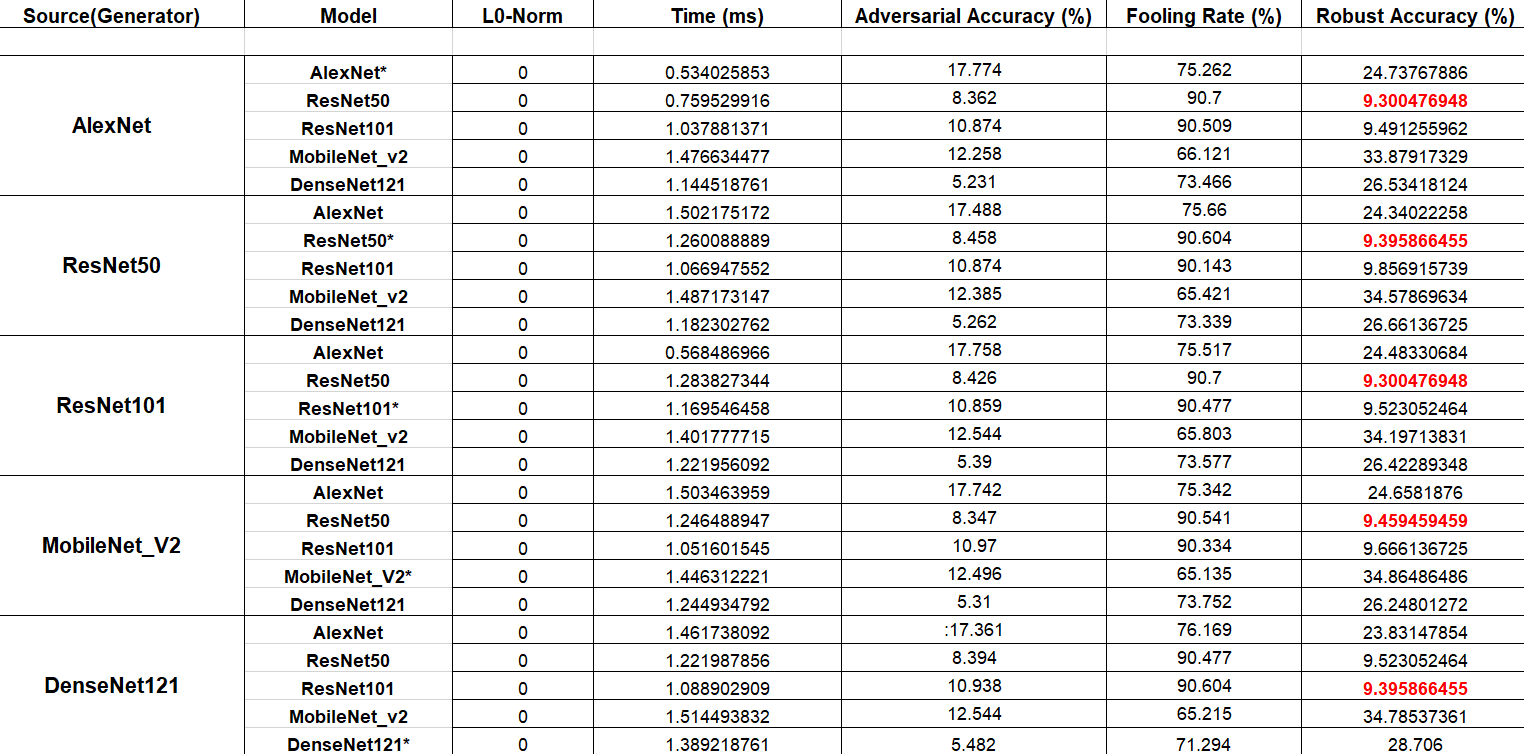
Confusion Matrix for PGD Attack on MobileNetV2

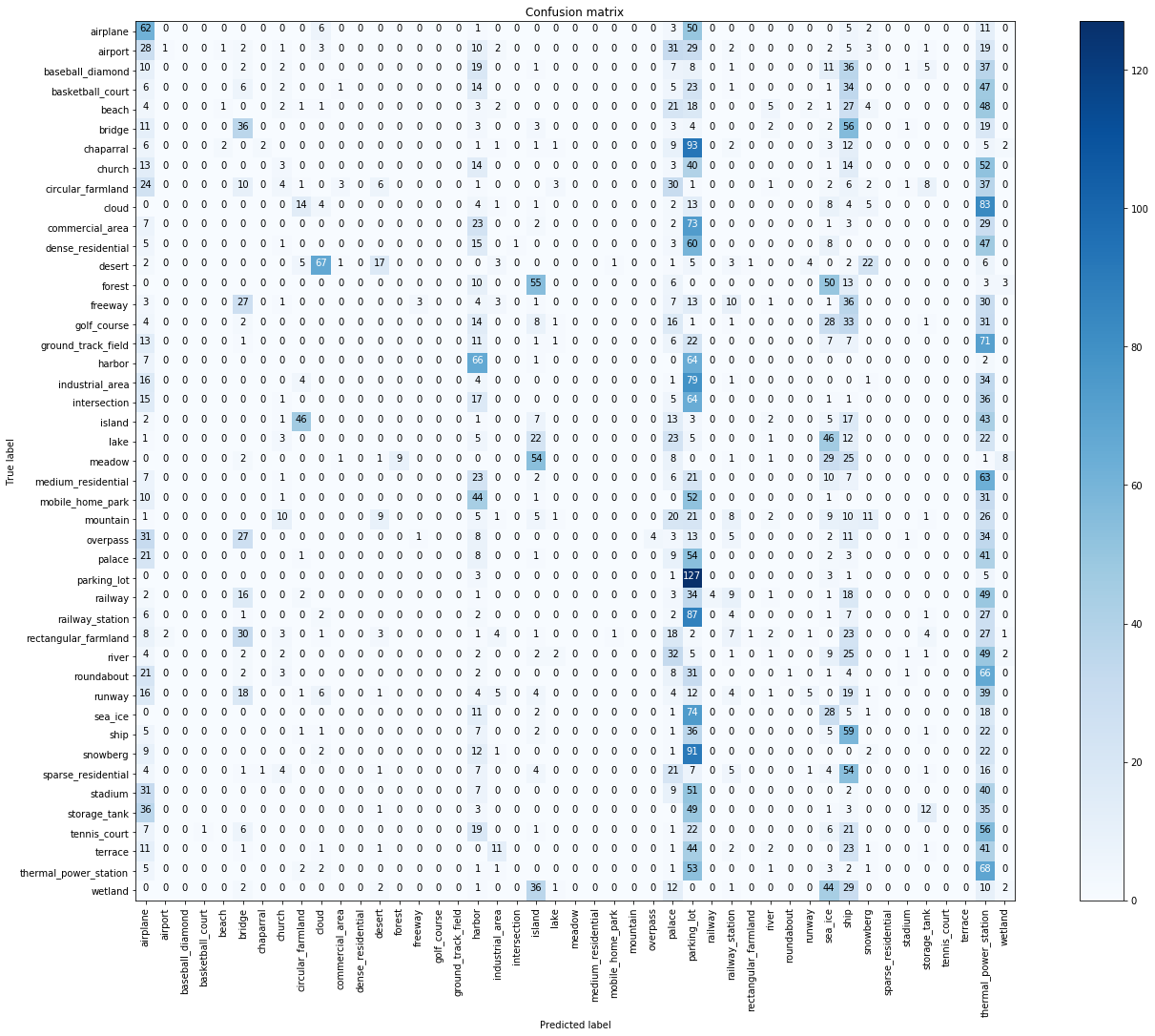
* Epsilon: 1.0



Confusion Matrix for L2-Projected Gradient Descent Attack on AlexNet

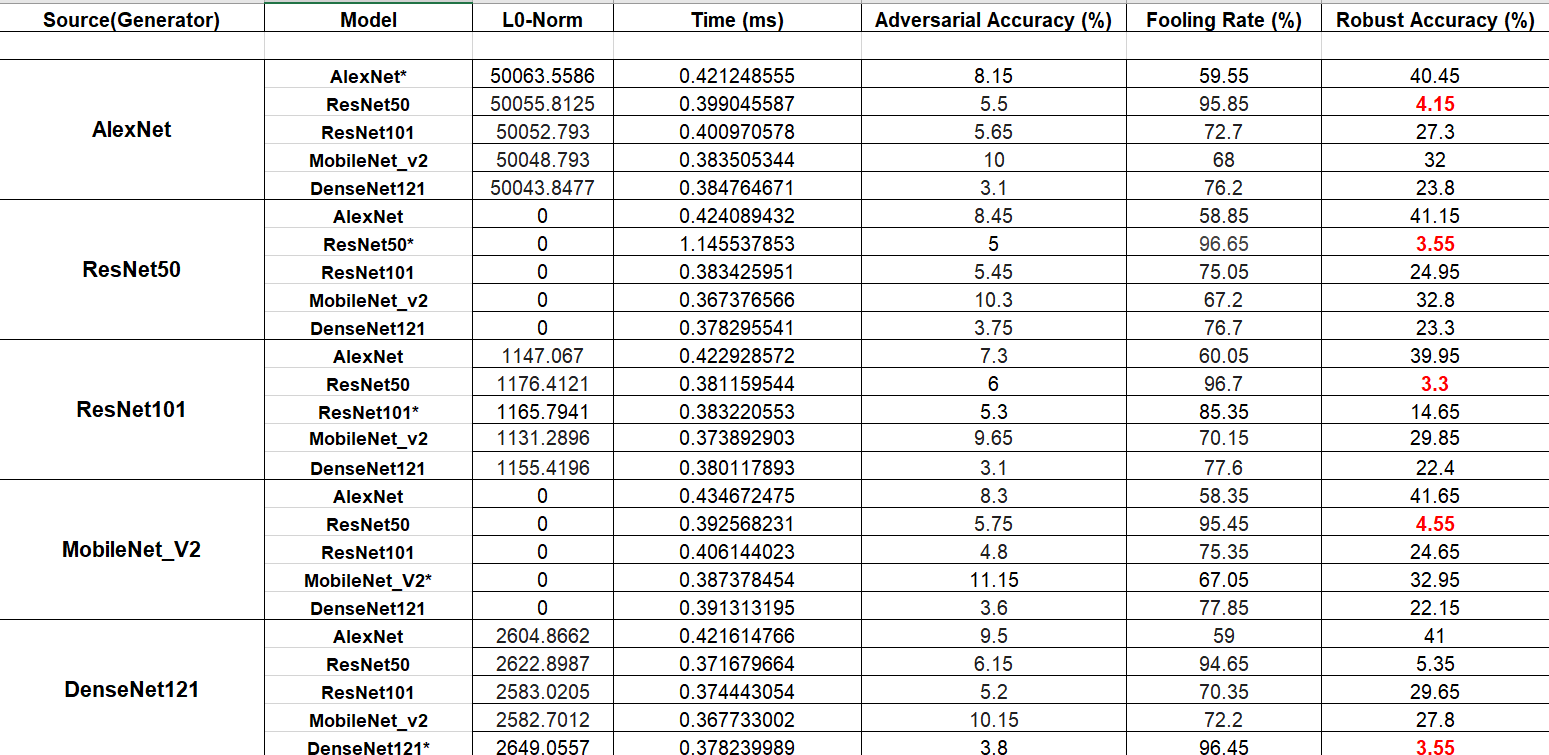
1. *Transferable Sparse Adversarial Attack (Black-Box Adversarial Attack)*
2. NWPU-RESIC45

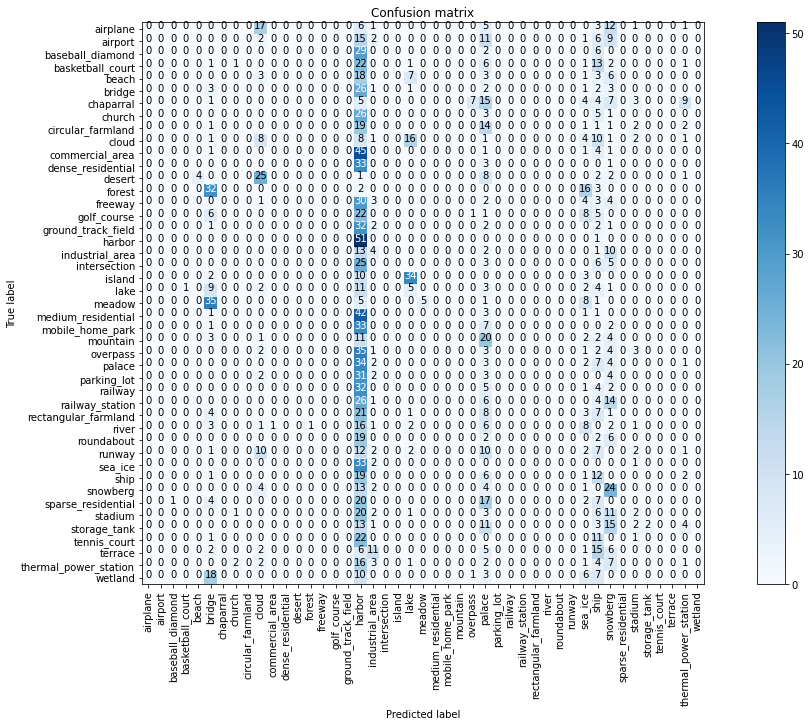
* Epsilon: 0.0005



Confusion Matrix for ResNet101 Generator Attack on ResNet50

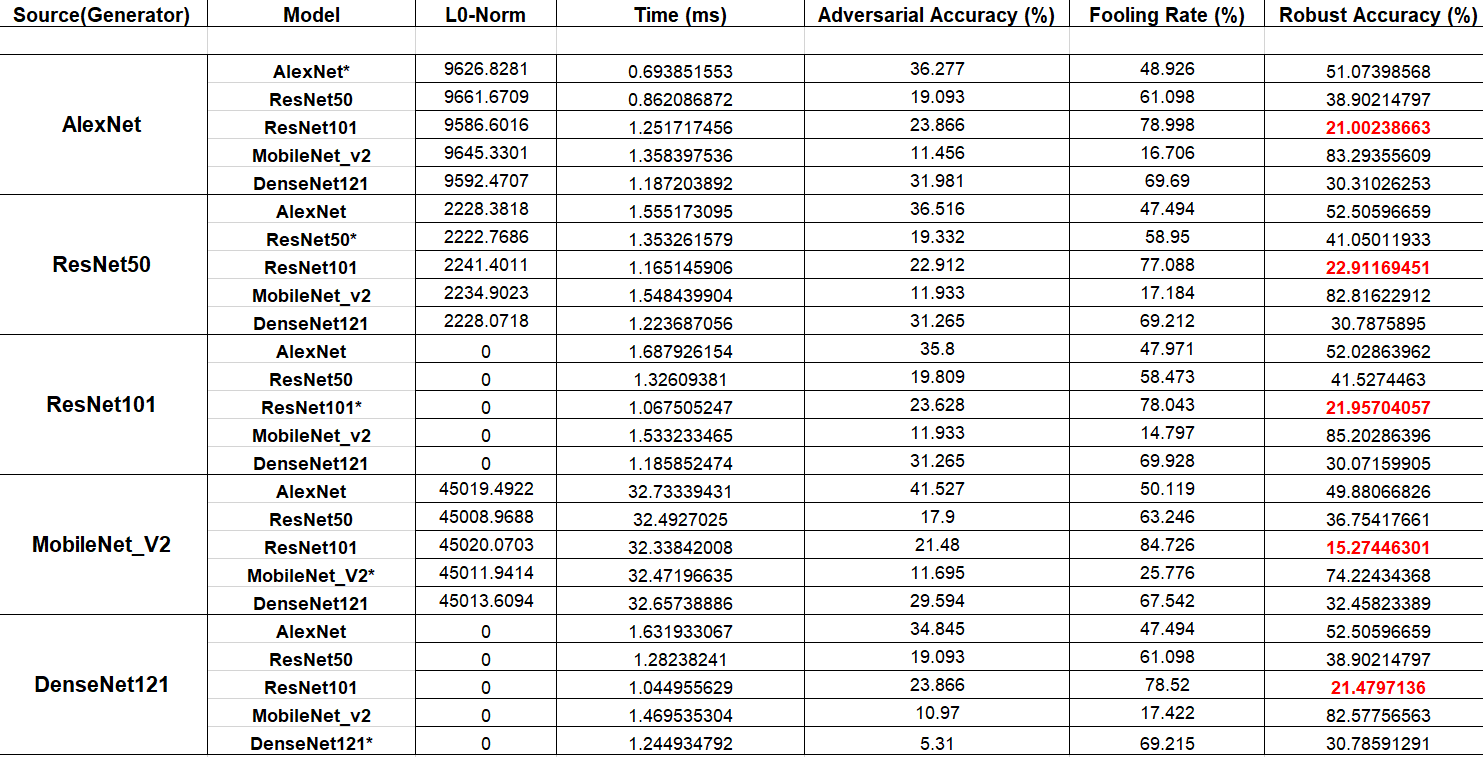
* Epsilon: 1.0

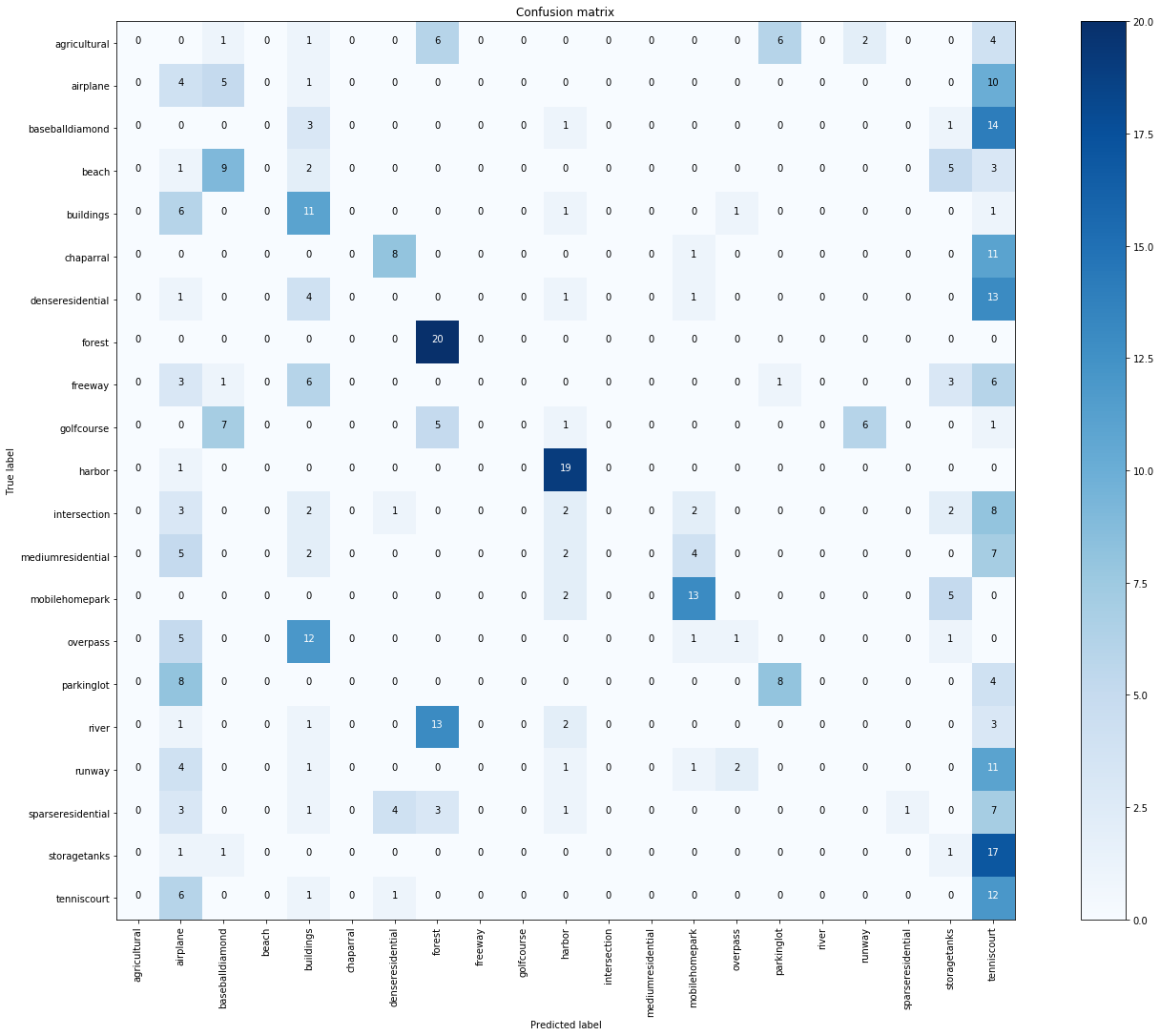




Confusion Matrix for ResNet101 Generator Attack on ResNet50

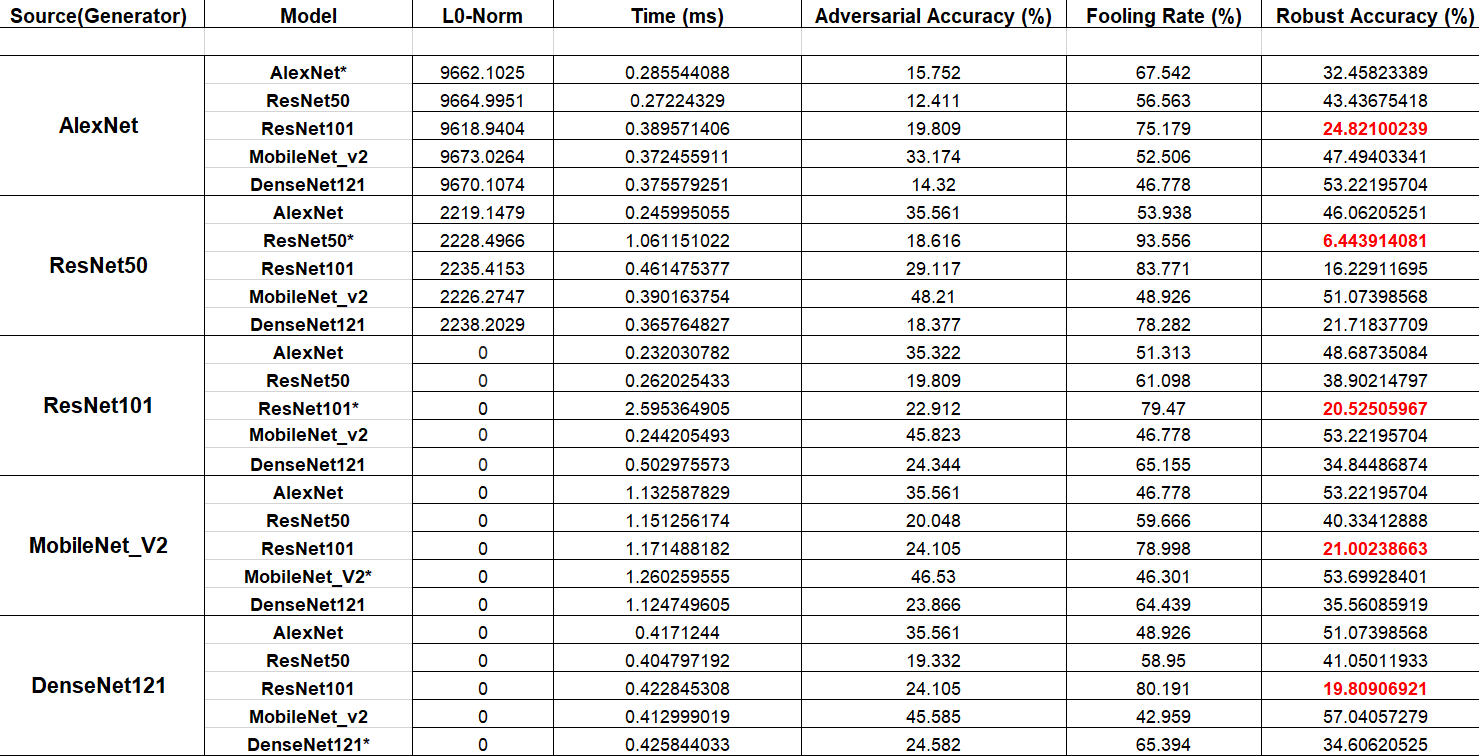
1. UC Merced Land Use Dataset

* Epsilon: 0.0005



Confusion Matrix for MobileNetV2 Generator Attack on ResNet101

* Epsilon: 1.0



# 

Confusion Matrix for ResNet50 Generator Attack on ResNet50

# Experiment Observations

# Following Observations were made on the basis of the experiments conducted in the above study:

# On NWPU-RESIC45, the highest drop in accuracy was observed when Virtual Adversarial Attack (75.23%) and Newton Fool Attack (44.67%) were deployed under eps= 0.0005 and eps=1.0 respectively. to create adversarial examples on the AlexNet and DenseNet respectively.

# When TSAA was executed on NWPU dataset, highest fooling rate was 9.30% when the generator was trained on ResNet101 architecture and the adversarial examples were created using the ResNet50 model under eps=0.0005 and 3.3% when the generator was trained on ResNet101 architecture and the adversarial examples were created using the ResNet50 model under eps=1.0

# On UC Merced Dataset, the highest drop in accuracy was observed when L2 Projected Gradient Descent Attack (4.01%) and PGD Attack (98.24%) were deployed under eps= 0.0005 and eps=1.0 respectively. to create adversarial examples on the AlexNet and MobileNetV2 respectively.

# When TSAA was executed on UC Merced dataset, highest fooling rate was 15.27% when the generator was trained on MobileNetV2 architecture and the adversarial examples were created using the ResNet101 model under eps=0.0005 and 6.44% when the generator was trained on ResNet50 architecture and the adversarial examples were created using the ResNet50 model under eps=1.0

# Experiment Discussion

# In this experiment, we verify that adversarial examples also exist in RSI scene classification systems. We find that the use of training datasets with different scales and different model structures has an impact on model vulnerability; even the same model exhibits different model vulnerabilities for different attack algorithms.

Adversarial examples of RSIs also have attack selectivity. From a geometric point of view, the RSI dataset is represented as a data point in the high-dimensional space in the CNN model, and the adversarial example is deviated from the boundary of the original cluster by adding small perturbations to the data points, which causes the models to misclassify it. Therefore, data points at a cluster boundary are more likely to be successfully attacked than data points near the center of the cluster. In addition, we find that most of the adversarial classes are beaches. Beaches might have special characteristics. We believe that this research can supply ideas for future research on security and defense methods concerning adversarial examples in RSI scene classification systems.

The research on adversarial examples of RSIs is still rare. Compared to natural images, adversarial examples of RSIs also have the same properties in some aspects. However, attack selectivity of adversarial examples is a property not explored in natural images. This property might be more helpful for understanding the correlation between ground objects. This study also has some limitations: (1) in the real scenario, adversarial examples of RSIs are usually placed with some physical objects, which would change the impact of the adversarial example problem [69]. (2) RSI has spectral and spatial properties. Different spatial resolutions would produce different adversarial examples, and these adversarial examples would have different properties [70]. In further research, we will explore these issues. Importantly, we will design defensive algorithms to solve the adversarial example problem.

# Conclusion and Future Work

In this article, we discuss the adversarial example problem in RSI scene classification for the first time. The experiment shows that adversarial examples are difficult for the human eye to recognize, and they allow the RSI scene classification system to achieve false results. This situation can cause serious problems for applications based on RSI scene classification systems.

Adversarial examples in the field of RSIs are a problem that should be addressed. In the field of RSI applications, many detection and segmentation tasks remain, such as building detection and land cover classification. The processing of these tasks is different from that of image classification. The adversarial example problems of these tasks require further study, and this topic is of great significance for the design of a secure and robust CNN-based remote sensing application system.

# Acknowledgment

This work was carried out in part using computing resources at the Indian Institute of Technology Hyderabad as a part of my summer internship project. I would like to sincerely acknowledge Prof. C. Krishna Mohan and his lab members for giving me this opportunity to work on the above problem

# Code

# The code for the above project can be found out on: [GitHub](https://github.com/Shorya-Sharma/Adversarial-Attacks-on-Aerial-Scene-Classification/tree/main)

# For accessing the experiment results and pre-trained weights files, please use the following drive link: [Google Drive](https://drive.google.com/drive/folders/1R3LFXK3ygZevZfVAs0bjCqF_w-3Ci0AU?usp=sharing)

# References

1. T. Lillesand, R. W. Kiefer, and J. Chipman, *Remote Sensing and Image Interpretation*. Hoboken, NJ, USA: Wiley, 2015.
2. J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, ‘‘Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning,’’ *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.
3. G. Forestier, A. Puissant, C. Wemmert, and P. Gançarski, ‘‘Knowledge- based region labeling for remote sensing image interpretation,’’ *Comput., Environ. Urban Syst.*, vol. 36, no. 5, pp. 470–480, Sep. 2012.
4. J. Zhang, ‘‘Multi-source remote sensing data fusion: Status and trends,’’

*Int. J. Image Data Fusion*, vol. 1, no. 1, pp. 5–24, Mar. 2010.

1. N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, ‘‘Deep learning classification of land cover and crop types using remote sensing data,’’ *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 778–782, May 2017.
2. K. Jia, Q. Li, Y. Tian, B. Wu, F. Zhang, and J. Meng, ‘‘Crop classifica- tion using multi-configuration SAR data in the North China plain,’’ *Int. J. Remote Sens.*, vol. 33, no. 1, pp. 170–183, Jan. 2012.
3. C. Conrad, S. Fritsch, J. Zeidler, G. Rücker, and S. Dech, ‘‘Per-field irrigated crop classification in arid Central Asia using SPOT and ASTER data,’’ *Remote Sens.*, vol. 2, no. 4, pp. 1035–1056, Apr. 2010.
4. C. Yuan, Y. Zhang, and Z. Liu, ‘‘A survey on technologies for automatic forest fire monitoring, detection, and fighting using unmanned aerial vehi- cles and remote sensing techniques,’’ *Can. J. Forest Res.*, vol. 45, no. 7, pp. 783–792, Jul. 2015.
5. J. C. White, N. C. Coops, M. A. Wulder, M. Vastaranta, T. Hilker, and

P. Tompalski, ‘‘Remote sensing technologies for enhancing forest inven- tories: A review,’’ *Can. J. Remote Sens.*, vol. 42, no. 5, pp. 619–641, Sep. 2016.

1. L. Tang and G. Shao, ‘‘Drone remote sensing for forestry research and practices,’’ *J. Forestry Res.*, vol. 26, no. 4, pp. 791–797, Dec. 2015.
2. G. J. Scott, M. R. England, W. A. Starms, R. A. Marcum, and C. H. Davis, ‘‘Training deep convolutional neural networks for Land–Cover classifica- tion of high-resolution imagery,’’ *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 4, pp. 549–553, Apr. 2017.
3. K. Jia, S. Liang, N. Zhang, X. Wei, X. Gu, X. Zhao, Y. Yao, and X. Xie, ‘‘Land cover classification of finer resolution remote sensing data inte- grating temporal features from time series coarser resolution data,’’ *ISPRS*

*J. Photogramm. Remote Sens.*, vol. 93, pp. 49–55, Jul. 2014.

1. M. Awrangjeb, M. Ravanbakhsh, and C. S. Fraser, ‘‘Automatic detection of residential buildings using LIDAR data and multispectral imagery,’’ *ISPRS*

*J. Photogramm. Remote Sens.*, vol. 65, no. 5, pp. 457–467, Sep. 2010.

1. M. Vakalopoulou, K. Karantzalos, N. Komodakis, and N. Paragios, ‘‘Build- ing detection in very high resolution multispectral data with deep learning features,’’ in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1873–1876.
2. A. Manno-Kovacs and T. Sziranyi, ‘‘Orientation-selective building detec- tion in aerial images,’’ *ISPRS J. Photogramm. Remote Sens.*, vol. 108, pp. 94–112, Oct. 2015.
3. G. Cheng, P. Zhou, and J. Han, ‘‘Learning rotation-invariant convo- lutional neural networks for object detection in VHR optical remote sensing images,’’ *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 12, pp. 7405–7415, Dec. 2016.
4. X. Li, W. Li, A. Middel, S. L. Harlan, A. J. Brazel, and B. L. Turner, ‘‘Remote sensing of the surface urban heat island and land architecture in Phoenix, Arizona: Combined effects of land composition and con- figuration and cadastral–demographic–economic factors,’’ *Remote Sens. Environ.*, vol. 174, pp. 233–243, Mar. 2016.
5. R. J. Shrivastava and J. L. Gebelein, ‘‘Land cover classification and economic assessment of citrus groves using remote sensing,’’ *ISPRS*

*J. Photogramm. Remote Sens.*, vol. 61, no. 5, pp. 341–353, Jan. 2007.

1. Y. Ke and L. J. Quackenbush, ‘‘A review of methods for automatic indi- vidual tree-crown detection and delineation from passive remote sensing,’’ *Int. J. Remote Sens.*, vol. 32, no. 17, pp. 4725–4747, Sep. 2011.
2. J. Hu, A. Razdan, J. C. Femiani, M. Cui, and P. Wonka, ‘‘Road net- work extraction and intersection detection from aerial images by tracking road footprints,’’ *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 12, pp. 4144–4157, Dec. 2007.
3. G. Fu, H. Zhao, C. Li, and L. Shi, ‘‘A method by improved circular projection matching of tracking twisty road from remote sensing imagery,’’ *Acta Geodaetica et Cartographica Sinica*, vol. 43, no. 7, pp. 724–730, 2014.
4. R. Crippen, ‘‘Calculating the vegetation index faster,’’ *Remote Sens. Envi- ron.*, vol. 34, no. 1, pp. 71–73, Oct. 1990.
5. S. K. McFEETERS, ‘‘The use of the normalized difference water index (NDWI) in the delineation of open water features,’’ *Int. J. Remote Sens.*, vol. 17, no. 7, pp. 1425–1432, May 1996.
6. X. Huang and L. Zhang, ‘‘Morphological building/shadow index for build- ing extraction from high-resolution imagery over urban areas,’’ *IEEE*

*J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 1, pp. 161–172, Feb. 2012.

1. J. Chen, M. Deng, X. Mei, T. Chen, Q. Shao, and L. Hong, ‘‘Optimal segmentation of a high-resolution remote-sensing image guided by area and boundary,’’ *Int. J. Remote Sens.*, vol. 35, no. 19, pp. 6914–6939, Oct. 2014.
2. G. Liasis and S. Stavrou, ‘‘Building extraction in satellite images using active contours and colour features,’’ *Int. J. Remote Sens.*, vol. 37, no. 5, pp. 1127–1153, Mar. 2016.
3. G. Cheng and J. Han, ‘‘A survey on object detection in optical remote sensing images,’’ *ISPRS J. Photogramm. Remote Sens.*, vol. 117, pp. 11–28, Jul. 2016.
4. Y. Li, H. Zhang, X. Xue, Y. Jiang, and Q. Shen, ‘‘Deep learning for remote sensing image classification: A survey,’’ *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 8, no. 6, p. e1264, Nov. 2018.
5. L. Ma, Y. Liu, X. Zhang, Y. Ye, G. Yin, and B. A. Johnson, ‘‘Deep learn- ing in remote sensing applications: A meta-analysis and review,’’ *ISPRS*

*J. Photogramm. Remote Sens.*, vol. 152, pp. 166–177, Jun. 2019.

1. L. Deng, ‘‘A tutorial survey of architectures, algorithms, and applications for deep learning,’’ *APSIPA Trans. Signal Inf. Process.*, vol. 3, pp. 1–29, Jan. 2014.
2. Y. Chen, Z. Lin, X. Zhao, G. Wang, and Y. Gu, ‘‘Deep learning-based classification of hyperspectral data,’’ *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 7, no. 6, pp. 2094–2107, Jun. 2014.
3. Y. Chen, H. Jiang, C. Li, X. Jia, and P. Ghamisi, ‘‘Deep feature extrac- tion and classification of hyperspectral images based on convolutional neural networks,’’ *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 6232–6251, Oct. 2016.
4. G. Cheng, C. Yang, X. Yao, L. Guo, and J. Han, ‘‘When deep learn- ing meets metric learning: Remote sensing image scene classification via learning discriminative CNNs,’’ *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 5, pp. 2811–2821, May 2018.
5. F. Zhang, B. Du, and L. Zhang, ‘‘Scene classification via a gradient boosting random convolutional network framework,’’ *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, Mar. 2016.
6. S. Chaib, H. Liu, Y. Gu, and H. Yao, ‘‘Deep feature fusion for VHR remote sensing scene classification,’’ *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 8, pp. 4775–4784, Aug. 2017.
7. N. Akhtar and A. Mian, ‘‘Threat of adversarial attacks on deep learning in computer vision: A survey,’’ *IEEE Access*, vol. 6, pp. 14410–14430, 2018.
8. I. J. Goodfellow, J. Shlens, and C. Szegedy, ‘‘Explaining and harnessing adversarial examples,’’ in *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*, San Diego, CA, USA, Y. Bengio and Y. LeCun, Eds. May 2015. [Online]. Available: <http://arxiv.org/abs/1412.6572>
9. A. Kurakin, I. Goodfellow, and S. Bengio, ‘‘Adversarial examples in the physical world,’’ 2016, *arXiv:1607.02533*. [Online]. Available: <http://arxiv.org/abs/1607.02533>
10. A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, ‘‘Synthesizing robust adversarial examples,’’ 2017, *arXiv:1707.07397*. [Online]. Available: <http://arxiv.org/abs/1707.07397>
11. C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan,
    1. Goodfellow, and R. Fergus, ‘‘Intriguing properties of neural networks,’’ 2013, *arXiv:1312.6199*. [Online]. Available: [http://arxiv.org/abs/1312.](http://arxiv.org/abs/1312) 6199
12. N. Carlini and D. Wagner, ‘‘Adversarial examples are not easily detected: Bypassing ten detection methods,’’ in *Proc. 10th ACM Workshop Artif. Intell. Secur. (AISec)*, 2017, pp. 3–14.