Siamese Network with Multi-Level Features for Patch-based Change Detection in Satellite Imagery

Faiz Ur Rahman[1], Bhavan Vasu[1], Jared Van Cor[2]
John Kerekes[2], Andreas Savakis[1]

[1]Vision and Image Processing Laboratory

[2]Digital Imaging and Remote Sensing Laboratory

Rochester Institute of Technology

Rochester, New York, USA

Outline

- Change Detection and Motivation
- CNN’s and Siamese Networks
- Simulation Dataset: DIRSIG
- Results
  - Using Fully Connected Decision Network
  - Use of Bootstrapping
  - Use of Euclidean Distance and Thresholding
- Conclusions

Change Detection

- Identifying changes of interest in each set of images is a fundamental task in computer vision with numerous applications like
  - Fault detection
  - Disaster management
  - Crop monitoring
  - Aerial surveying
- We investigated changes in helicopter detection (present/not present)

Fig1

Fig1:Vector Illustration of GIS Spatial Data Layers Concept for Business Analysis, Geographic Information System, Icons Design, Liner Style. Source: Adobe Stock Images

Change Detection: Algebraic Approaches

Earlier attempts to detect changes varied in complexity from some basic algebraic methods to several machine learning methods. Some basic algebraic methods were:

- Image differencing [3]
- Connection vector analysis [4]
- Ratioing [5]

But the above approaches yield a high false positive due to illumination changes were developed for single source imagery.

---

[3] D. Muchoney and B. Haack, "Change detection for monitoring forest defoliation," Photogrammetric Engineering and Remote Sensing, vol. 60, pp. 1243–1251, 1994.

[4] E. F. Lambin, "Change detection at multiple temporal scales: seasonal and annual variations in landscape variables," Photogrammetric Engineering and Remote Sensing, vol. 62, no. 8, pp. 931–938, 1996.

[5] A. Prakash and R. P. Gupta, "Land-use mapping and change detection in a coal mining area - a case study in the Jharia coalfield, India," International Journal of Remote Sensing, vol. 19, no. 3, pp. 391–410, 1998.

Change Detection: Advanced Approaches

Some techniques from machine learning were also adopted to reduce data redundancy between bands and predict pixels including:

- Principal Component Analysis (PCA) [6]
- Tasseled Cap transformations [7]
- Gramm-Schmidt [8]
- Chi-square [9]
- Image regression to predict pixels [10]

These were limited as all changes were detected not just changes of interest.

[6] G. Byrne, P. Crapper, and K. Mayo, "Monitoring land-cover change by principal component analysis of multi-temporal landsat data," Remote sensing of Environment, vol. 10, no. 3, pp. 175–184, 1980.

[7] K. C. Seto, C. Woodcock, C. Song, X. Huang, J. Lu, and R. Kaufmann, "Monitoring land-use change in the pearl river delta using Landsat," International Journal of Remote Sensing, vol. 23, pp. 1985–2004, 2002.

[8] J. B. Collins and C. E. Woodcock, "An assessment of several linear change detection techniques for mapping forest mortality using multitemporal Landsat data," Remote sensing of environment, vol. 56, pp. 66–77, 1996.

[9] M. K. Ridd and J. Liu, "A comparison of four algorithms for change detection in an urban environment," Remote sensing of environment, vol. 63, no. 2, pp. 95–100, 1998

[10] A. Singh, "Spectral separability of tropical forest cover classes," International Journal of Remote Sensing, vol. 8, no. 7, pp. 971–979, 1987.

Deep CNN as a Feature Extractor

Deep Learning Approaches: The work in [1] proposed using a deep convolutional neural network as a feature extractor followed by a stage of segmentation and thresholding.

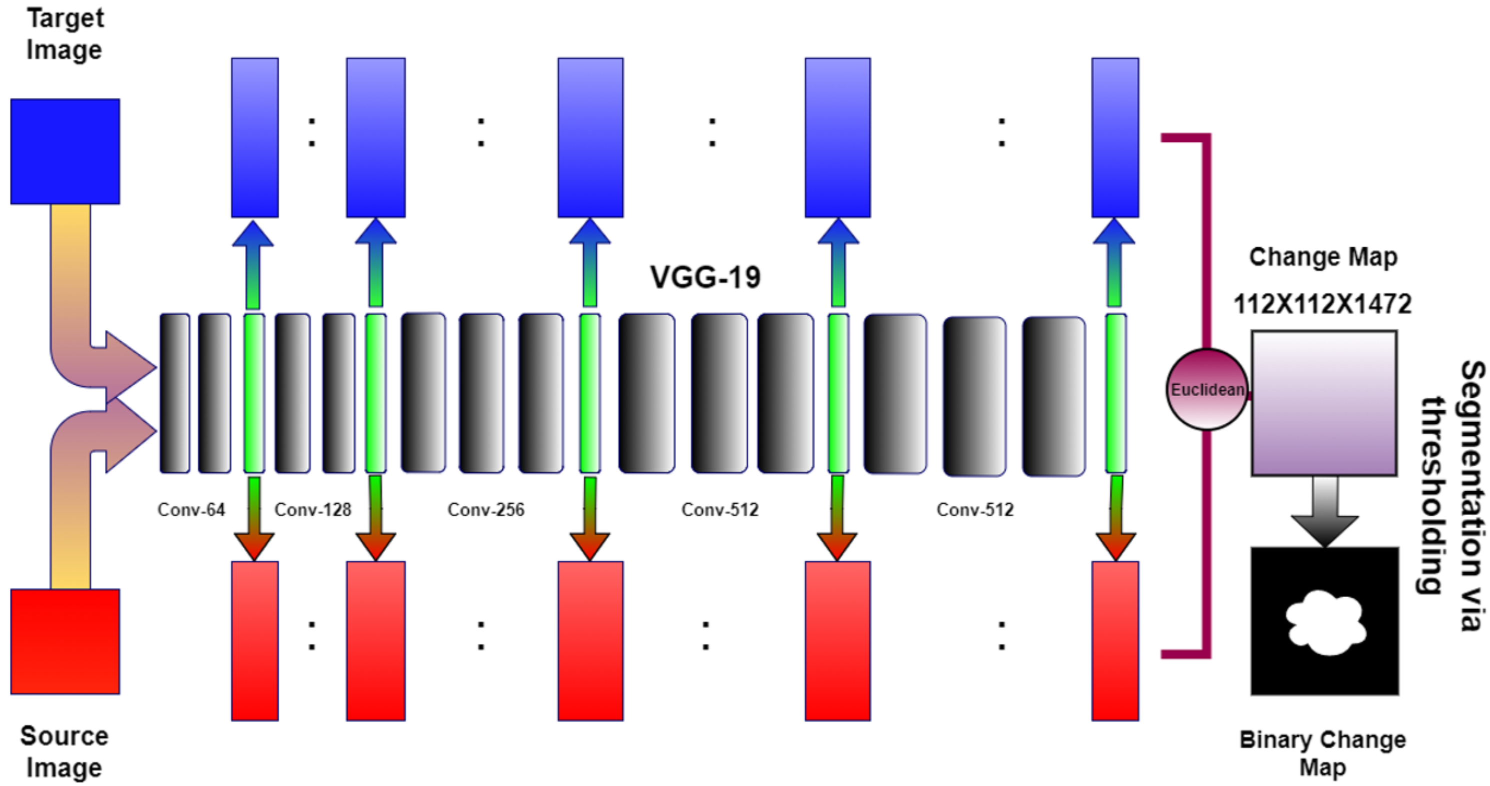


Fig 2: Block diagram of convolutional neural network features based change detection

[1] A. M. El Amin, Q. Liu, and Y. Wang, "Convolutional neural network features based change detection in satellite images," in First International Workshop on Pattern Recognition International Society for Optics and Photonics, 2016.

Siamese Architectures

- A Siamese Architecture has two identical channels, that are usually based on standard CNN architectures. Each of the channels processes different images.
- Potential applications of Siamese networks include
  - Person Re-Identification
  - Object Tracking
  - Change Detection
- A Siamese CNN was used for change detection with threshold segmentation
  - Y. Zhan, K. Fu, M. Yan, X. Sun, H. Wang, and X. Qiu, "Change detection based on deep Siamese convolutional network for optical aerial images," IEEE Geoscience and Remote Sensing Letters, 2017.
- A fully convolutional Siamese architecture was used for change detection of hyperspectral images
  - R. Daudt, B. Le Saux, A. Boulch, "Fully Convolutional Siamese Networks for Change Detection," IEEE Int. Conf. Image Processing (ICIP), 2018.

Siamese Network Architecture

- Our Siamese network has two identical convolutional networks that merge into a common decision network.
- The convolutional networks are VGG16 architectures pre-trained on ImageNet.
- Both VGG16 networks share their trainable parameters.
- The decision network is trained to detect changes between the Target and Reference Images.

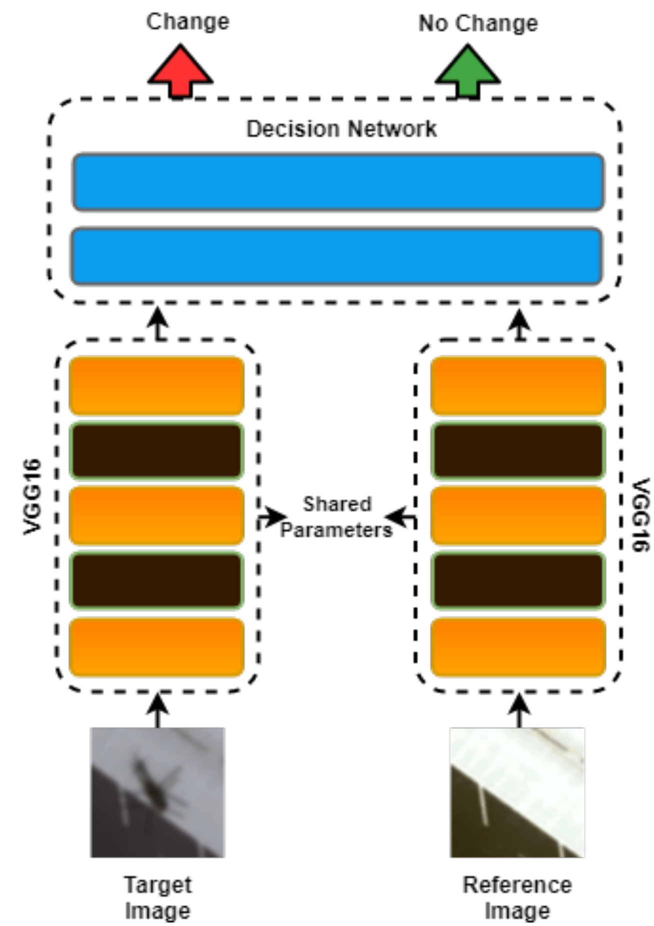


Fig 4: Siamese Neural Network Architecture with Decision Network

Multi-Level Feature Concatenation

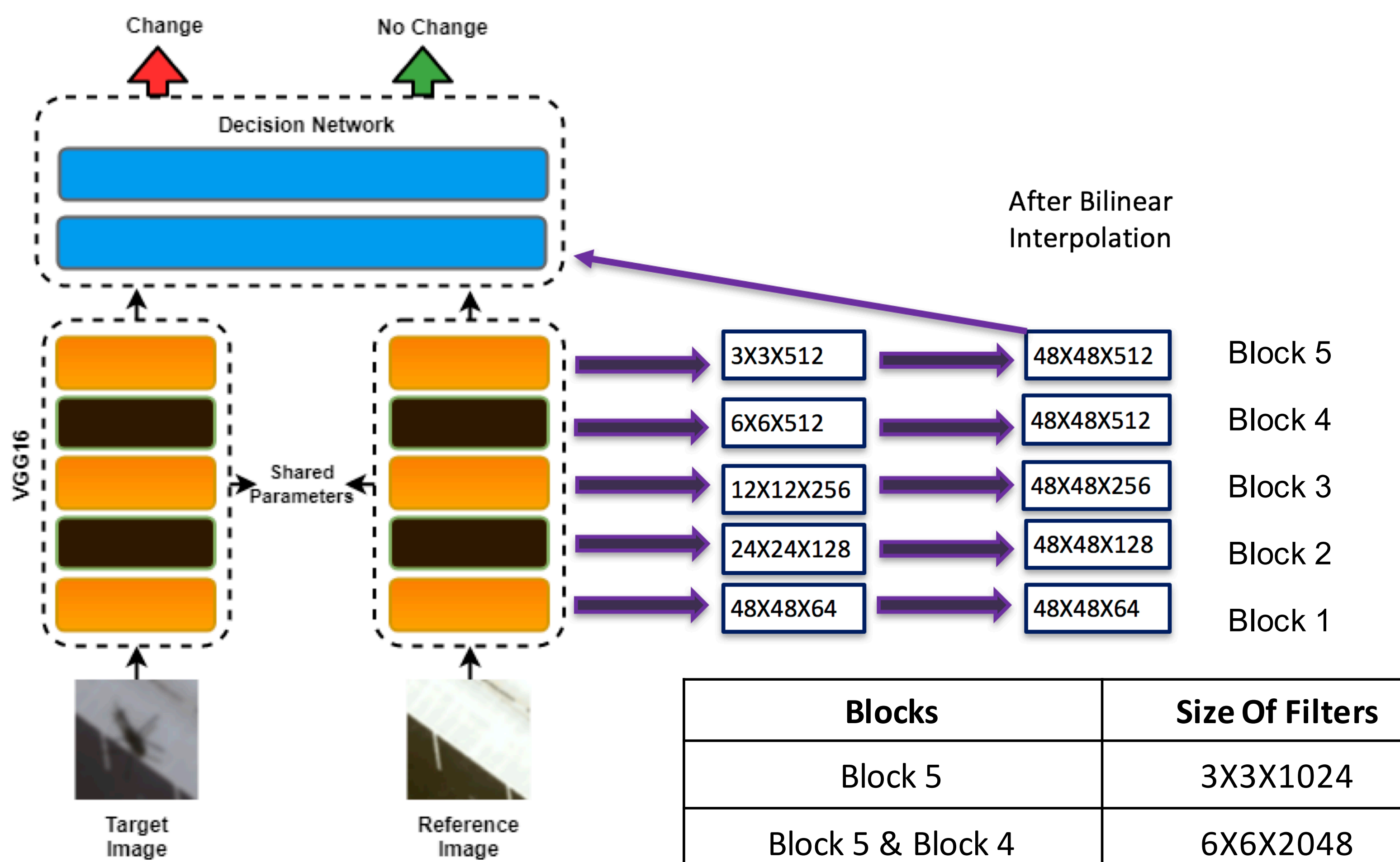


Fig 5: Siamese Neural Network multi-level feature concatenation.

Blocks	**Size Of Filters**
Block 5	3X3X1024
Block 5 & Block 4	6X6X2048
Block 5, Block 4 & Block 3	12X12X2560

Decision Network for Siamese Network

- Decision Network consists of two Fully Connected (FC) layers.
- This tensor variable consists of 1024 tensors, 512x3x3 from each network (for Block 5).
- The first FC layer takes as input the outputs of the two Siamese VGG16 blocks.
- The second FC layer outputs correspond to two classes: **Change** and **No Change**.

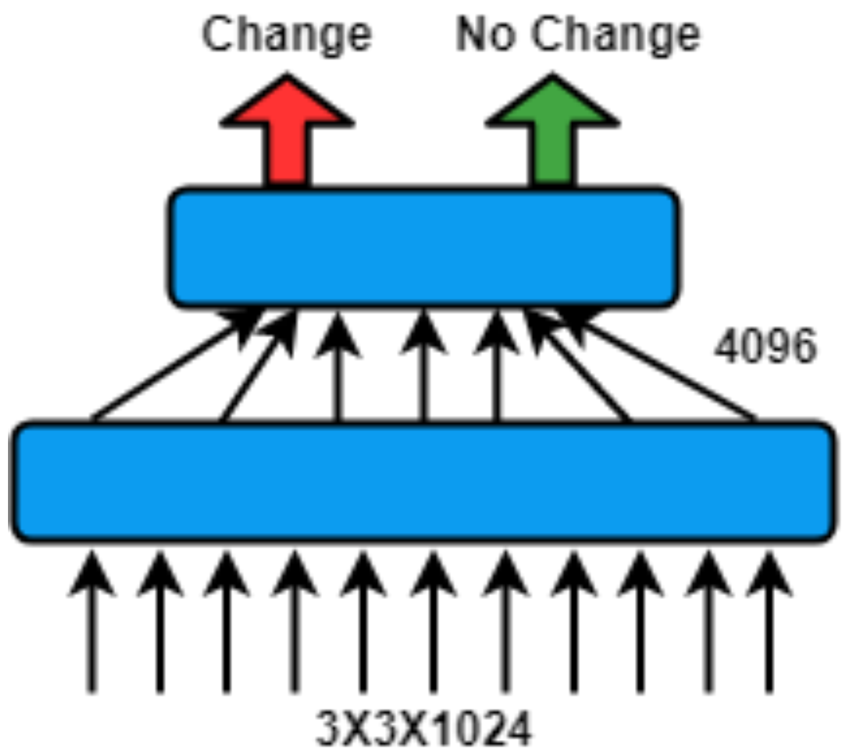


Fig 6: Decision network architecture

Decision Network Training

- The convolution network (Siamese channel) is used to extract features that are stored for training the decision network.
- We interpolate the feature maps to 48 X 48, this enables us to stack features from different layers.
- Extracted features from different VGG blocks are concatenated before passing them to the fully connected layer.
- We experiment with VGG features from just Block 5, concatenating Block 4 and Block 5 and this process is continued until the we have features from Block 1, Block 2, Block 3, Block 4 & Block 5.

Digital Imaging and Remote Sensing Image Generation (DIRSIG)

- RIT developed DIRSIG is leading tool for scene and image chain simulations
- Common uses include:
  - Sensor/system performance evaluation
  - Algorithm training and evaluation
  - Rapid parametric analysis
- Image Modalities
  - Visible through thermal infrared (0.4 - 20.0 microns)
  - Passive Sensing
    - Broad-band, multi-spectral (MS), hyperspectral (HS) imaging
    - Polarization (usually limited by material properties)
  - Active Laser sensing
    - Topographic LADAR and atmospheric/gas LIDAR
- Instruments
  - Single pixel, 1D arrays and 2D arrays
  - Filter, diffraction/refraction or interferogram-based
  - Photon collection
- Platforms
  - Ground, air or space on static or moving platforms
- Phenomenology
  - Passive thermodynamics, secondary light sources, dynamic scene content (moving geometry), volume radiometry (plumes, water), etc.

MegaScene Tile #1
- 5,000+ objects
- 500+ million facets
- 1.6 km^2 (0.6 mi^2)

Change Detection Dataset

- The DIRSIG simulation environment was used to create our training and testing datasets.
- Images of helicopters and backgrounds were simulated to be representative of pan-sharpened DigitalGlobe WorldView-2 imagery.
- Image chips for Target and Reference images of size 80 × 80 were generated under different backgrounds and illumination, with a variety of spectral and structural properties.
- We made sure that our data samples contained enough variation to allow the network to learn changes related to the presence of helicopters and ignore illumination changes.

Example Simulated Image Chip Pairs

Image Pair 1&2

Image Pair 3&4

Image Pair 5&6

Image Pair 7&8

Our dataset consisted of

- 11,700 image pairs for training.
- 5,000 image pairs for testing.
- Variations in illumination for target/no target pairs

Decision Network Training (Block 5)

- Decision network trained on features from Block 5.
- Training accuracy 95.6%.

Decision network was trained for **80** epochs with a batch size of **150** and learning rate of **0.01** using **SGD** loss on a TITAN GTX 960. Same hyper parameters are used for further experiments.

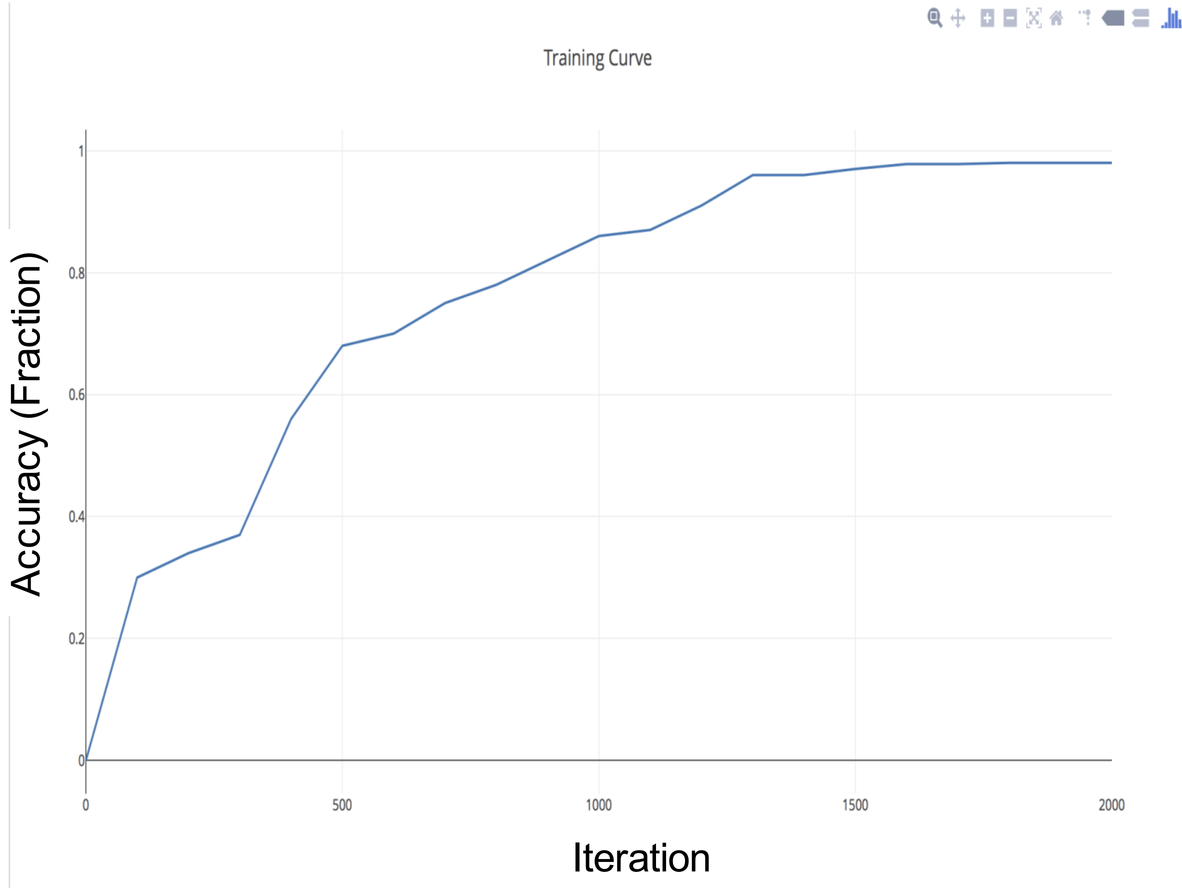


Fig 8: Training curve for decision network trained with Block 5 features.

Example Results (Block 5 Features)

Ground Truth: No Change Change Change No Change

Target Image

Reference Image

Output: No Change Change Change Change

Fig 9: Results for change detection using features from Block 5 to train decision network.

Training Curve (Block 4 & Block 5)

- Decision network trained on combined features from Block 4 and Block 5.
- Learning rate **0.01** with **SGD** Loss.
- Training Accuracy is 96.7%.

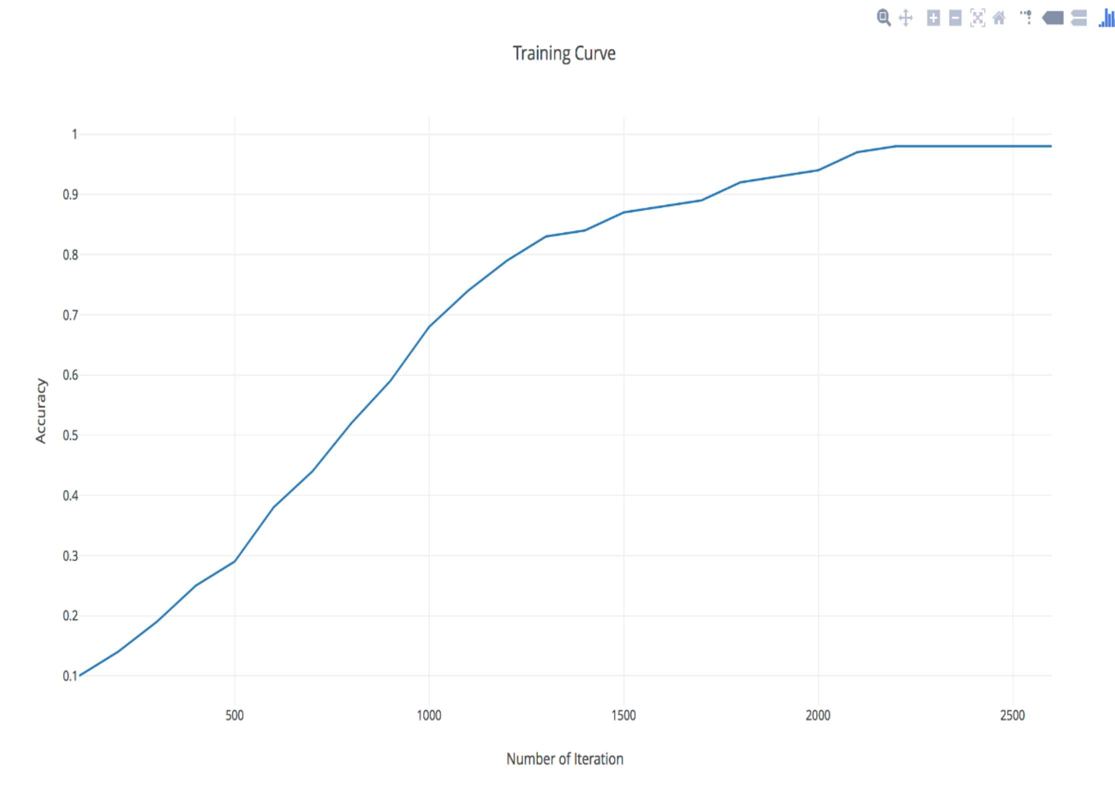


Fig 10: Training curve for decision network trained with Block 4 and Block 5 features.

Example Results (Block 5 & Block 4 Features)

Ground Truth: No Change | Change | No Change | No Change

Target Image

Reference Image

Output: No Change | Change | No Change | Change

Fig 11: Results for change detection using features from block 4 and block 5 to train decision network.

Results with Fully Connected Decision Network

Test Data Confusion Matrices

Block 5

Prediction→ Truth ↓	Change	NO Change
Change	4873	633
NO Change	127	4367

Block 5 & Block 4

Prediction→ Truth ↓	Change	NO Change
Change	4951	411
NO Change	49	4589

Accuracy

Number Of blocks	Test Accuracy (%)
Block 5	92.4
Block 5 & Block 4	**95.9**
Block 5, Block 4 & Block 3	93.5
Block 5, Block 4, Block 3 & Block 2	89.1
Block 5, Block 4, Block 3,Block 2 & Block 1	84.8

Bootstrapping

- Bootstrapping of image chips to increase the number of true positives detected.
- Bootstrapping is performed by re-training the final decision layer with all false negatives and false positives.

Accuracy

Features Used	Test Accuracy (%)
Block 5 & Block 4 (Bootstrapping)	96.5

20

Results with Bootstrapping

Ground Truth:	No Change	Change	No Change	Change

Target Image

Reference Image

Output:	No Change	Change	No Change	No Change

Fig 12: Results for change detection using features from block 4 and block 5 with bootstrapping

Siamese Network with Euclidian Distance Comparison

- Euclidian distance is a simple yet effective distance measure for comparing samples *p* and *q* in n-dim feature space.

$$\sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}.$$

- Euclidian Distance above a Threshold T=0.76 is used to detect a Change, else No Change.
- This architecture does not require supervision (training) other than the selection of the threshold parameter.

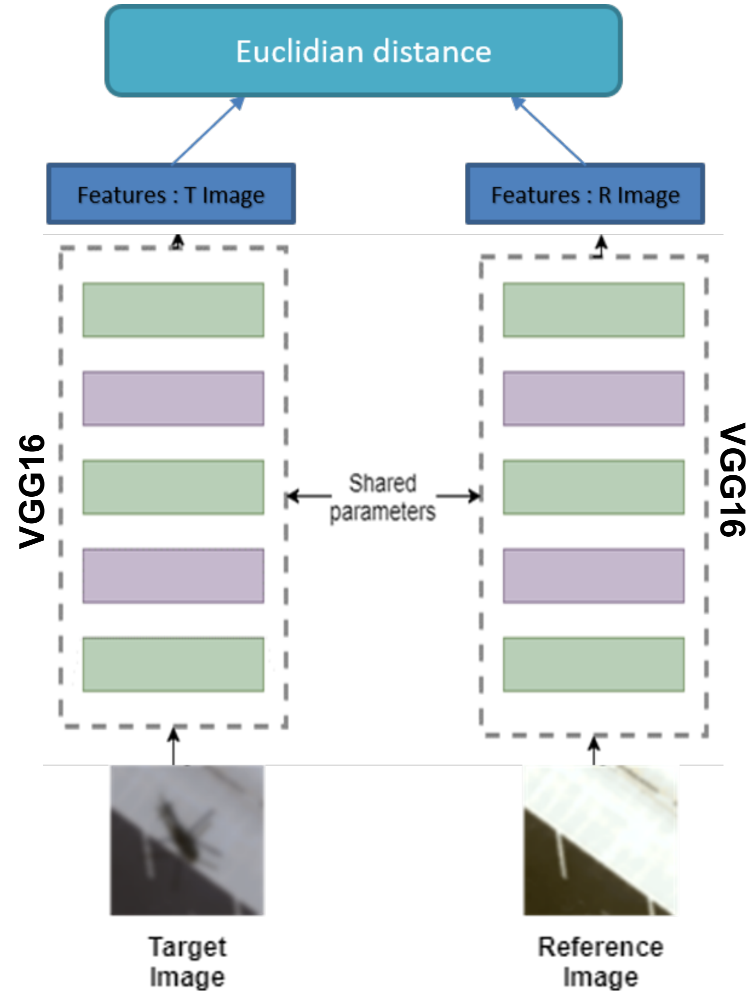


Fig 13: Siamese network with Euclidean distance

Results Using Euclidean Distance

Features	Test Accuracy (%)	Comments
Block 5	91.8	
Block 5 & 4	**93.4**	**Gives better results with more features**
Block 5 & 4 & 3	90.6	Starts giving false too many false positives
Block 5 & 4 & 3 & 2	88.5	Too many false positives
Block 5 & 4 Euclidean (threshold=0.52)	86.9	All true Positives with many false positives

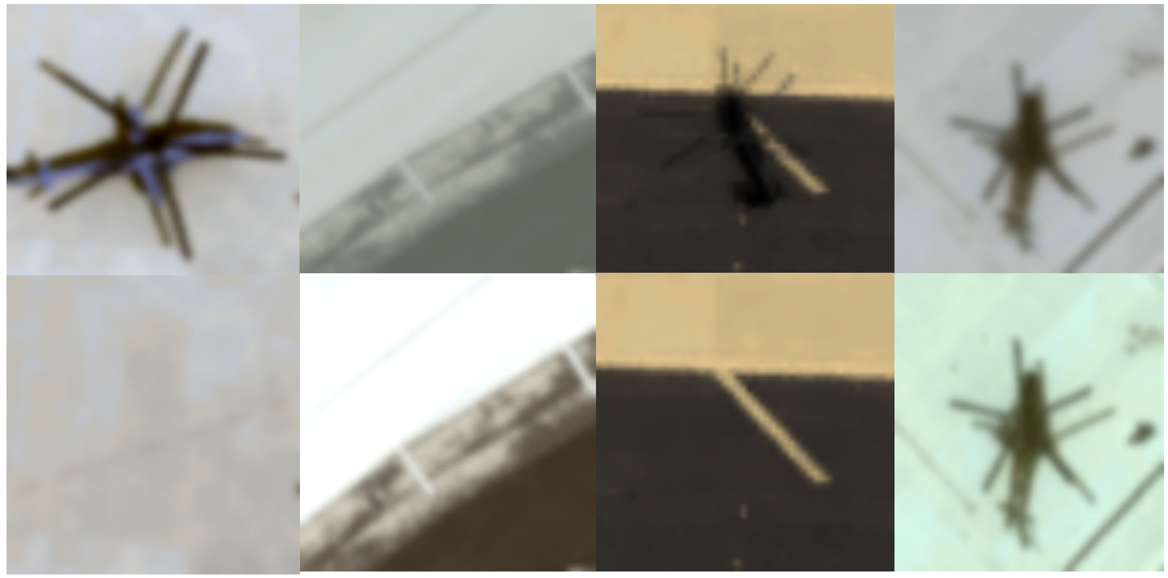
- We observed that the algorithm generally gets confused when helicopter is on the tarmac
- We experimented with decreasing the threshold for the Euclidean distance to 0.52 (from 0.76) to detect all the true positives, but this resulted in many false positives.

Results using Euclidean (Block 5 & Block 4)

Ground Truth:	Change	No Change	Change	No Change

Target Image

Source Image



Output:	Change	No Change	Change	Change

Fig 14: Results for change detection using features from Block 5 and Block 4 with Euclidean distance.

Conclusions

- We proposed a Siamese architecture that can detect changes in the presence of helicopters between two satellite image pairs.
- Experiments with multilevel VGG features and a decision network showed that the features from Block 4 and Block 5, when combined, were the most useful at detecting changes.
- Bootstrapping was seen to aid in increasing the true positive rate of our Siamese architecture.
- The Siamese architecture with a trained decision network outperformed a Siamese network with simple thresholding of Euclidean distance.