Detection of Metadata Tampering through Discrepancy between Image Content and Metadata using Multi-task Deep Learning

Bor-Chun Chen*
University of Maryland
College Park
sirius@umd.edu

Pallabi Ghosh*
University of Maryland
College Park
pallabig@umd.edu

Vlad I. Morariu
University of Maryland
College Park
morariu@umd.edu

Larry S. Davis
University of Maryland
College Park
lsd@umiacs.umd.edu

Abstract

Image content or metadata editing software availability and ease of use has resulted in a high demand for automatic image tamper detection algorithms. Most previous work has focused on detection of tampered image content, whereas we develop techniques to detect metadata tampering in outdoor images using sun altitude angle and other meteorological information like temperature, humidity and weather, which can be observed in most outdoor image scenes. To train and evaluate our technique, we create a large dataset of outdoor images labeled with sun altitude angle and other meteorological data (AMOS+M2), which to our knowledge, is the largest publicly available dataset of its kind. Using this dataset, we train separate regression models for sun altitude angle, temperature and humidity and a classification model for weather to detect any discrepancy between image content and its metadata. Finally, a joint multi-task network for these four features shows a relative improvement of 15.5% compared to each of them individually. We include a detailed analysis for using these networks to detect various types of modification to location and time information in image metadata.

1. Introduction

Tampered image metadata is frequently encountered in image forensics. Ease of metadata access and modification using simple EXIF tools has resulted in tampered images that are difficult to detect, except in very special cases or after rigorous expert investigations. Our goal in this paper is to automate this process and reduce the effort required by experts.

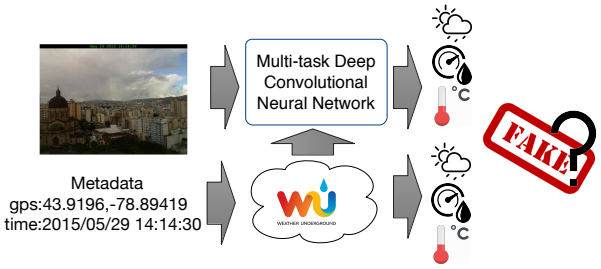

Figure 1. Given an image, we first detect information such as sun altitude angle, temperature, humidity and weather conditions using multi-task deep learning. We then compare the inferred properties to the same information collected from the Internet based on image metadata to detect if there is any tampering.

One of the areas where image metadata authenticity is very important is legal cases where an image is shown as evidence of a certain activity at a certain time. The timestamp of the image cannot be trusted just on its own as it is easily modified. It needs to be corroborated by some additional information in the image if available. For example, in the Duke Lacrosse case [1], the timestamp of one of the images matched with the timestamp of one of the player's watch. We develop automatic techniques to perform similar types of analysis in outdoor images using meteorological information.

We focus on image location and timestamp tamper detection, as these two are the most important factors in the image metadata. Existing research has focused on checking the validity of the location information by matching image content against a large-scale image database such as Google

*These authors contributed equally to this work.

street view images using content-based image retrieval techniques. However, this only works well with very few locations having distinct features such as tourist landmarks.

Although it is hard to directly infer location and time from the image content, recent research has shown that advances in machine learning have enabled reasonably accurate prediction of meteorological information directly from image content [4, 15, 13, 16]. Therefore, we utilize the sun altitude angle and other historical meteorological information such as temperature, humidity and weather — all available on the web — to detect image metadata tampering. Our goal is to infer meteorological properties separately, directly from image content and then compare them to the same properties obtained from historical weather databases at the time and location specified in the image metadata. We expect that, unless the image metadata was carefully tampered with to ensure consistency with weather patterns, metadata tampering will lead to inconsistencies that can be detected by our proposed algorithm.

To train and evaluate our approach, we first collect a large-scale dataset (AMOS+M2) with images, metadata (i.e., timestamps and GPS location), as well as sun altitude angle and meteorological information based on the already existing AMOS [8] database and the Weather Underground Internet API [3]. We then use AMOS+M2 to learn different convolutional models for prediction. In order to utilize the correlation between different sources of information, we further propose a joint model based on multi-task learning, which predicts all of the features simultaneously.

While there has been some work in this area, our novelty lies in the fact that our test and training data comes from different web cameras, and our research includes the results of applying these models to image forensics. Also, by combining different networks using multi-task learning, we are able to further improve the prediction accuracy. Figure 1 shows the overview of our system.

The main contributions of this paper include: (1) analyzing the use of sun altitude angle and meteorological information for image content vs. metadata discrepancy detection; (2) exploiting the benefit of multi-task learning on meteorological information and sun angle prediction; (3) constructing a large-scale dataset called AMOS+M2 containing more than 500,000 outdoor images labeled with the above mentioned information and the metadata.

2. Related Works

There has been a large amount of research in the field of digital image forensics. Sencar *et al*. [14] provide a survey of the different available digital image forensics techniques. The survey includes methods based on image source identification, synthetic image identification, and detection of image tampering. Most of the tampering detection techniques perform statistical analysis of the different kinds of variations in the observed signals after tampering.

Although there have been many successes in detecting tampering from image content, existing techniques generally do not deal with image metadata tampering. Kakar *et al*. [9] is one of the few that have addressed this problem. However, instead of using only sun angle for detection, we combine other meteorological information available on the Internet and apply multi-task deep learning to further improve accuracy.

Other related works have focused on prediction of sun angle or other meteorological information: Lalonde *et al*. [12] use mathematical models based on sun illumination, shadow length and direction and shading of vertical surfaces to estimate the sun position and illumination, and others have also investigated similar approaches [4, 16, 7, 13]. Recently, Volokitin *et al*. [15] applied deep convolutional neural networks for temperature and time prediction. However, none of these methods utilize different meteorological information with multi-task deep learning. Some of them only train and test on images from the same webcam. Our goal is to learn a general model that can be applied to any outdoor image, captured by any camera, at any location or time for metadata tampering detection.

3. Sun angle and meteorological information prediction

We use convolutional neural network (CNN) models to predict sun angle and meteorological information. We experiment with two variants of convolutional models for our prediction tasks: AlexNet [11] and ResNet-50 [6] . AlexNet contains five convolutional layers followed by three fully connected layers, while ResNet-50 contains 49 convolutional layers with residual connections followed by one average pooling layer. We use AlexNet to experiment with different loss functions (mean squared and mean absolute losses) due to the advantage of its training speed and use ResNet-50 to train our final model to obtain better prediction results.

3.1. CNN for temperature, humidity, and sun angle regression

To use CNN for regression tasks, we first replace the last layer of the CNN with a single output using a distance based loss function. Since the outputs of our regression models should always lie in certain ranges (e.g. zero to ninety degrees for sun angle), we use a sigmoid or an extra ReLU-like nonlinear layer to clip the output from both sides before the final loss layer; but they improve performance only in some cases whereas decrease performance in others. We also weight the training loss based on the probability distribution of the ground truth labels and call these the weighted regression models. This helps to give

more importance to the examples that are less common in the training set and tries to solve the problem that the dataset is not uniformly distributed. Finally, we train the network with our AMOS+M2 dataset.

3.2. CNN for weather classification

For weather classification, we train a classification CNN with our AMOS+M2 dataset. We first separate our training data into four different classes: sunny, cloudy, rainy, and snowy. Since our training set is highly unbalanced, as sunny and cloudy images together take around 85% of the training set, directly training the network would cause the model to be biased toward sunny and cloudy. To address this issue, we apply data oversampling with augmentation: for each image class, we first oversample the images to make each class have roughly the same size, and then we apply data augmentation to each oversampled image by first randomly resizing and keeping the smallest side of the image between 256 to 512 pixels. We then randomly crop the image down to 227×227 and randomly apply a left-right flip to the image. Finally, we adopt the softmax cross entropy loss function to optimize the network parameters. In order to reduce the training time, we initialize the weights of our network to a model pretrained on ImageNet dataset.

3.3. CNN with joint multi-task learning

Since all of the meteorological information we use is correlated, it is natural to wonder if one model can benefit from the others. Therefore, we use multi-task learning to learn a joint model that can predict all the meteorological information at the same time. This is achieved by weight sharing on all the regression and classification networks with a joint loss function. We adopt the same network architecture, ResNet-50, for all tasks so that we can share the weights crossing all four tasks. Since the four different tasks have different output ranges, we first normalize each output to zero mean and one standard deviation so that each loss function will be the same scale. Let $X = [x_1, ..., x_7]$ be the output of our joint network, and $Y = [y_1, ..., y_7]$ the value of the meteorological information, where $(y_1, ..., y_4)$ is a one-hot encoding vector of weather condition, and y_5, y_6, y_7 represent sun altitude angle, temperature, and humidity respectively. We minimize the following joint loss function:

$$L(X,Y) = -\sum_{i=1}^{4} \log y_i p(x_i) + \sum_{i=5}^{7} ||x_i - \frac{(y_i - \mu_i)}{\sigma_i}||^2, \quad (1)$$

where, $\mu_5, \mu_6, \mu_7, \sigma_5, \sigma_6, \sigma_7$ represent the mean and standard deviation of sun altitude angle, temperature, and humidity in the training set. $p(x_i)$ represents the probability of the i_{th} class being the correct weather computed by the softmax function. We train this joint model with an initial learning rate of 0.0002 and a mini-batch size of 256 images using Adam optimizer [10].

Figure 2. Example of boundary images where the sun altitude angle changes from negative to positive. We use these boundary images for manually verifying the camera location. If the camera geographic location is incorrect, the calculated sun altitude angles will be incorrect, and therefore, it is less likely that such day/night boundary can be identified in the image content. So we check boundary images to filter out cameras with incorrect location annotations.

4. Metadata and meteorological information outdoor scenes dataset (AMOS+M2)

In order to train our model for metadata tampering detection, we construct a large-scale image dataset called AMOS+M2.

4.1. Data collection

We collect images from Archive of Many Outdoor Scenes (AMOS), an archive of images collected from Internet webcams since 2006. Each image in AMOS contains a timestamp and a camera ID, and each camera may contain its location annotated by the AMOS user as well as the IP location of the webcam. Note that the timestamp associated with any image is mostly correct because it is automatically generated by the system, but the location of the camera can be missing or incorrect.

In order to verify the location of the cameras, we first compute the distance between the location derived from the camera IP address and the annotated location and filter out cameras when this distance is greater than 100 miles. We then compute the sun altitude angle for each image based on the timestamps and the annotated camera location using Pysolar [2] and detect the sunrise and sunset boundary, where sun angle changes between positive and negative numbers. If the location is correct, we should be able to visually see large illumination differences between these boundary images as shown in Figure 2. We manually check these boundary images to remove cameras with incorrect GPS locations.

After manual verification, we use the Weather Underground API [3] to collect all the relevant meteorological information including temperature, humidity and weather conditions based on the locations and the timestamps of the images.

Dataset	# of locations	# of images	Metadata	Meteorological information	Sun angle
Weather Image Dataset [13]	N/A	10K	N	weather	N
Multi-class Weather Image [16]	N/A	20K	N	weather	N
Glasner*et al*. [5]	10	6K	Y	temperature	N
Time of the Year Dataset [15]	10	23K	Y	temperature	N
AMOS+M2 (Ours)	638	500K	Y	weather, temperature, humidity	Y

Table 1. Comparison between AMOS+M2 with other existing datasets. AMOS+M2 contains more images from different locations; with more detailed meteorological information as well as sun altitude angles.

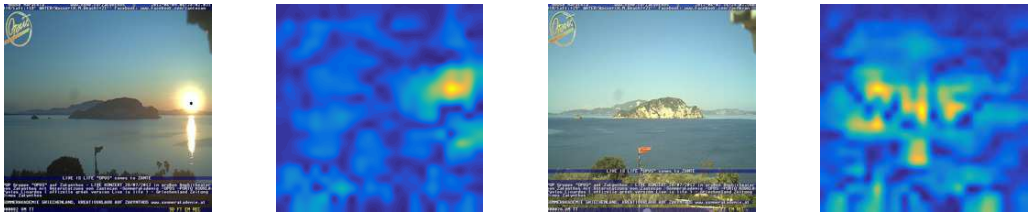

Figure 3. Heat maps of absolute difference in output sun altitude angle predictions when small portions of the images are occluded. The two images are from the same webcam at different times. In the first set, we can see that the network gives importance to the sun if it is visible in the image. In the second set the network gives importance to the reflective rock surfaces.

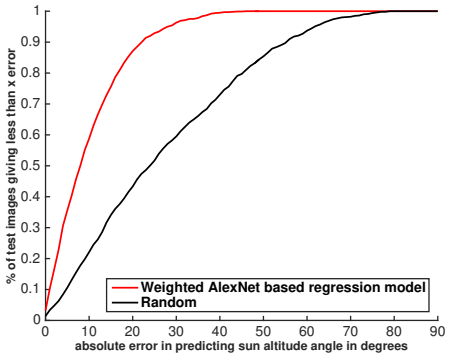

Figure 4. The x-axis in the figure is the absolute error in the prediction of sun altitude angle and the y-axis is the percentage of test images giving error less than or equal to the corresponding x value. The higher the area under the curve, the better is the result. For the sun altitude angle test set, the model resulted in 55% of test images with less than or equal to 10°error and about 85% of test images with less than or equal to 20°error.

4.2. Dataset statistics

We obtain 638 cameras from AMOS with verified locations. We randomly select 538 cameras for training and the remaining 100 cameras are used for validation. For each camera in the training set, we randomly select around 1,000 images taken in 2016 to construct a training set of 500,000 images; for each camera in the testing set, we randomly select 10 images taken in 2016 to construct a test set of 1,000 images. Table 1 shows the dataset statistics compared to related works. Compared to existing datasets, AMOS+M2 contains more images from multiple locations, and more detailed meteorological information, as well as sun altitude angles, enabling us to effectively train our convolutional models.

The AMOS+M2 dataset with the images and corresponding meteorological data and metadata will be made publicly available.

5. Experiments on meteorological information and sun altitude angle prediction

5.1. Sun altitude angle regression

The performance of an AlexNet based L2 regression model for sun altitude angle is shown in Figure 4. The x axis in the figure is the absolute error in the prediction of sun altitude angle and the y axis is the percentage of test images giving error less than or equal to the corresponding x value.

Figure 4 shows that almost 55% of the images yield less than 10°error and about 85% of images give less than 20°error for the weighted regression model. The RMS sun angle prediction error for this model is 13.70°. On the other hand, the Resnet based model gives an RMSE of 11.31°.

To gain insight into the internal representation of the model, we visualize the heat maps of absolute difference in output predictions when we occlude small portions in the image. The results are shown in Figure 3. These images are from the same webcam taken at different times of the day. The heatmap shows which area has the most impact in

Figure 5. Ground truth vs the predicted temperature values for different scenes. The temperature model can predict temperatures even at night, which is not possible by the sun altitude angle model.

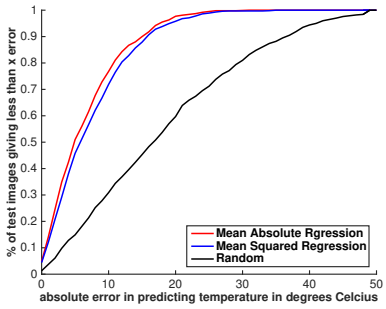

Figure 6. The x axis in the figure is the error in the prediction of temperature and the y axis is the percentage of test images giving error less than or equal to the corresponding x value. So about 80% of images have less than 10°C error and about 45% gives less than 5°C error for the mean absolute regression model. Also the mean absolute regression model performs better than the mean squared regression model.

determining the output sun altitude angle. When the sun is present in the first image, the model gives importance to that portion of the image. On the other hand, it gives importance to the reflective rock surface in the second image.

5.2. Temperature regression

We perform temperature prediction using an AlexNet based regression model with a mean absolute loss layer. The average temperature error is 8.94°C and the Pearson correlation between the ground truth and predicted temperatures is 0.7339. For the ResNet based model, the RMS error reduces to 7.45°C for the L2 loss based model. Figure5 shows the ground truth and predicted temperature values from two different images.

Figure6 and 7 show that mean absolute loss performs better than mean squared loss. Figure6 shows that about 45% of images give less than 5°C error and almost 80% give less than 10°C error for mean absolute regression. Figure 7 plots the variation of average error with the actual ground truth label. The flatter or more uniform the curve, the better are the results. As we can see, mean absolute regression works better than mean squared regression. The

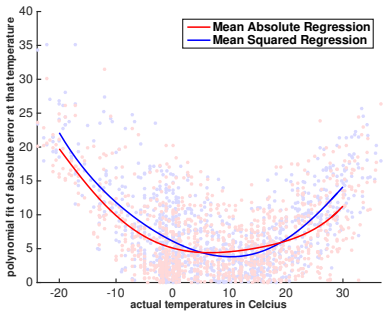

Figure 7. Polynomial fit to the error distribution vs. the ground truth temperature labels. The flatter the curve, the more uniform the error distribution across the output values. This shows that mean absolute regression performs better than the mean squared regression.

Pearson correlation coefficient for mean absolute regression is 0.7339, whereas for mean squared regression is 0.6689. The RMSE for mean absolute regression is 8.94°C whereas the RMSE for mean square regression is 9.83°C.

5.3. Humidity regression

We find that although it is hard to infer the exact percentage of humidity from the image, there are usually some weather related visual cues that indicate the range of the humidity in the scene. Figure 8 shows examples of images that predict as low humidity (i.e. lass than 30 percent) and high humidity (i.e. greater than 85 percent). The numbers under the images are the regression results and the numbers in parentheses are the ground-truth humidity percentages. As shown in Figure 8, low humidity images in the first row are associated with clear skies; while in the second row there can be rain, cloud, and snow indicating that the humidity values are high. Our regression network based on Alex Net achieves an average root mean square error (RMSE) of 18.42% whereas the Resnet based model achieves RMSE of 15.33%. Although the RMSE compared to the error in sun altitude angle and temperature regression is high, as shown in the following sections, our joint multi-task model can still benefit from the humidity information, which further improves the accuracy of metadata tampering detection.

5.4. Weather condition classification

Figure 9 shows the confusion matrix and some example classification results. The labels under the images are the output of the classifier and the labels in parentheses are the ground-truth labels. The red border indicates missclassifications. As shown in Figure 9, the classifier tends to classify rainy and snowy images as cloudy. This is because when it is raining or snowing, the sky looks cloudy as well. On the other hand, sometimes right after rain or snow, the road will look wet or covered with snow, which

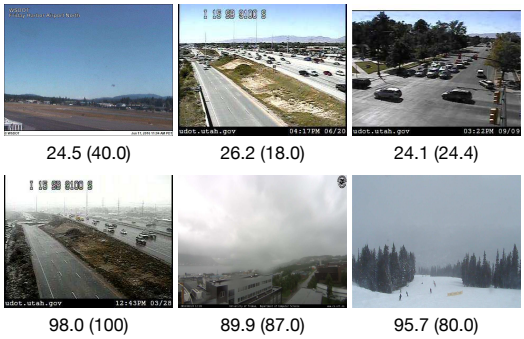

Figure 8. Example result of humidity regression. The number under each image is the predicted humidity and the number in the parentheses is the ground-truth. Top row: images predicted as having low humidity. Bottom row: images predicted as having high humidity. Although it is hard to predict the exact percentage of humidity from an image, there are usually some visual cues indicating the humidity range.

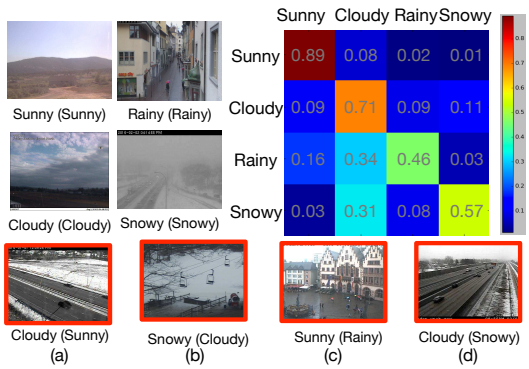

Figure 9. Example results and confusion matrix for weather condition classification. Red borders indicate misclassification. Rainy and snowy are prone to be misclassified as cloudy because the sky in each image is cloudy as well. (a) Sunny images misclassified as cloudy because the sky, which is an important cue for sunny images, is not visible. (b) Cloudy image misclassified as snowy because the snow covers a huge percentage of the image. (c) Rainy image misclassified as sunny because of the bright sky. (d) Snowy image misclassified as cloudy because the snow on the highway is mostly removed. Overall, our model can achieve 28.3% classification error rate.

is why it is harder to separate these classes. Our classifier achieves 23.9% classification error rate on the test set after 100K training iteration.

5.5. Joint multi-task learning

Table 2 compares the classification error rate and regression RMSE on four different tasks with models learned separately and jointly with multi-task learning. All models are trained with the same network structure using the same hyper-parameters with 100K training steps. As shown in the Table, all four tasks benefit from a joint model, with weather classification enjoying the highest relative improvement. This is probably because weather conditions are highly related to all three other tasks. After joint multi-task learning, we can achieve an RMSE of 10.81, 6.9, 15.09 for sun altitude angle, temperature, and humidity regression and an error rate of 23.9 for weather condition classification. In order to further analyze the benefit of multi-task learning for weather classification, we train three other models using one of the meteorological information sources (sun altitude angle, temperature, or humidity) as well as the weather condition as input labels. The results are shown in Table 3. Each type of meteorological information can slightly help with the weather classification, and the best performance is achieved by utilizing all of the meteorological information, which demonstrates the benefit of multi-task learning in meteorological information prediction.

Task	Single	Joint	Rel. Improv.
Sun Angle (RMSE)	11.31	**10.81**	4.4%
Temperature (RMSE)	7.45	**6.90**	7.4%
Humidity (RMSE)	15.33	**15.09**	1.6%
Weather (ERR)	28.30	**23.90**	15.5%

Table 2. The RMSE and classification error rate of the individual models and the joint model. Joint multi-task learning can improve the results for all four tasks and yields the most significant improvement for the weather classification because the weather is highly related to the other three sources of information.

Model	Error Rate
Weather	28.30
Weather, sun angle	27.30
Weather, temperature	27.80
Weather, humidity	27.50
All	**23.90**

Table 3. Weather classification error rate, combining meteorological information and sun altitude angle. Each slightly helps to reduce the classification error rate, and best performance is achieved by combining all the information, which demonstrates the effect of multi-task learning.

6. Experiments on metadata tampering detection

To analyze the effectiveness of meteorological information on tamper detection, we generate different tampered datasets by changing the timestamps or the GPS locations on the test images. We use ROC curves and Area under ROC curves (AUC) as our performance metrics. In the rest of this section, we discuss the results of tampering detection on different types of tampered test sets.

Month (AUC)	Angle	Humidity	Temp.	Weather
1	53.5%	62.5%	63.5%	**67.7%**
2	61.5%	71.4%	68.7%	**72.5%**
3	67.9%	72.3%	**80.7%**	71.1%
4	76.9%	74.9%	**84.5%**	74.1%
5	81.6%	72.2%	**84.6%**	73.7%
6	83.8%	73.6%	**85.1%**	71.6%

Table 4. AUC on time tampered data with large time tampering, in the order of multiple months. The weather model yields the best performance when the tampered time is one to two months from the ground truth because other information only changes slightly during a short period of time. The temperature model achieves the best performance when the tampered time is three to six months off from ground truth due to seasonal temperature changes. Sun altitude angle prediction yields better performance when the tampered time is further from ground truth because the sun position changes for the same time of the day throughout different seasons.

6.1. Time metadata tampering detection

We construct the time tampered dataset by changing the timestamps on half of the test images to create positive samples (i.e. tampered) while the rest of the test images maintain their authentic timestamps and serve as negative samples. The three types of time tampered datasets are constructed by changing the timestamps in the test images with different month, day, and hour variances respectively. We then use the absolute difference of the sun altitude angle, humidity, and temperature between the output of our model and the meteorological information downloaded from the Internet, as well as the weather probability score output to compute the ROC curve and the AUC percentage.

Table 4 shows the AUC using different types of meteorological information, on the time tampered dataset with tampering variation in months. As shown in the table, when the time difference is one to two months, the weather model has the best performance in detecting inconsistency. This is because the change in sun altitude angle, humidity, and the temperature is small and our model has a hard time perceiving differences in these properties. On the other hand, the weather classifier can better separate different weather conditions happening in different months. When the time difference is three to six months, the temperature model has the best performance, because there is seasonal change and temperature exhibits large differences, which can be detected by our model. Sun altitude angle performs better when the tampering time is larger because the sun angle at the same time of the day will change more with greater variation in months. The ROC curves for the temperature model based monthly time tamper detector are shown in Figure 10(a).

Table 5 shows the AUC on time tampering dataset with tamper quantity variation in days ranging from one to six days. The overall performance is worse than Table 4 because these meteorological measures exhibit less change during shorter intervals. The weather model again has the best performance overall and humidity has the second best.

Day (AUC)	Angle	Humidity	Temp.	Weather
1	50.3%	60.1%	54.7%	**67.5%**
2	50.3%	61.6%	57.1%	**70.1%**
3	49.7%	64.8%	56.4%	**69.5%**
4	50.2%	63.9%	56.0%	**69.6%**
5	49.6%	64.7%	57.9%	**69.2%**
6	50.2%	64.8%	59.3%	**69.9%**

Table 5. AUC on time tampered data with time tampering in the order of multiple days. The weather model achieves the best performance compared to other models. This is because all other meteorological information and sun altitude angle only has little change during short periods and it is hard to detect the difference.

Hour (AUC)	Angle	Humidity	Temp.	Weather
1	**58.6%**	53.1%	51.2%	53.8%
2	**65.3%**	56.5%	53.5%	53.8%
3	**68.6%**	59.2%	54.3%	56.5%
4	**71.9%**	60.6%	54.5%	57.8%
5	**74.8%**	61.4%	56.4%	58.2%
6	**73.7%**	63.1%	56.6%	58.7%

Table 6. AUC on time tampered data with time tampering in the order of multiple hours. The sun altitude angle model yields the best performance because the sun altitude angle changes throughout the day while the weather usually varies little in a day.

Table 6 shows the AUC on the time tamper dataset with tampering in hours ranging from one to six hours. As shown in the table, sun altitude angle model has the best performance and the performance increases as the number of hours increase. This is because weather usually does not change too much during the same day, while sun angle will keep changing throughout the day. The ROC curve for the sun angle based hourly time tampering detection model is as shown in Figure 10 (b).

Since each of these models performs best when detecting different types of tampering, we combine them all with late fusion by adding the normalized scores from each model. Figure 10 (c) shows the ROC curve with a tampered dataset that randomly changes the timestamps within a range from one hour to one year. By combining all the models, we leverage the strength of each model and achieve better performance on tamper detection.

6.2. Location metadata tampering detection

We construct two location tampered test sets by changing the latitude and the longitude of the image metadata respectively. Figure 11 shows the AUC of tampering detection on

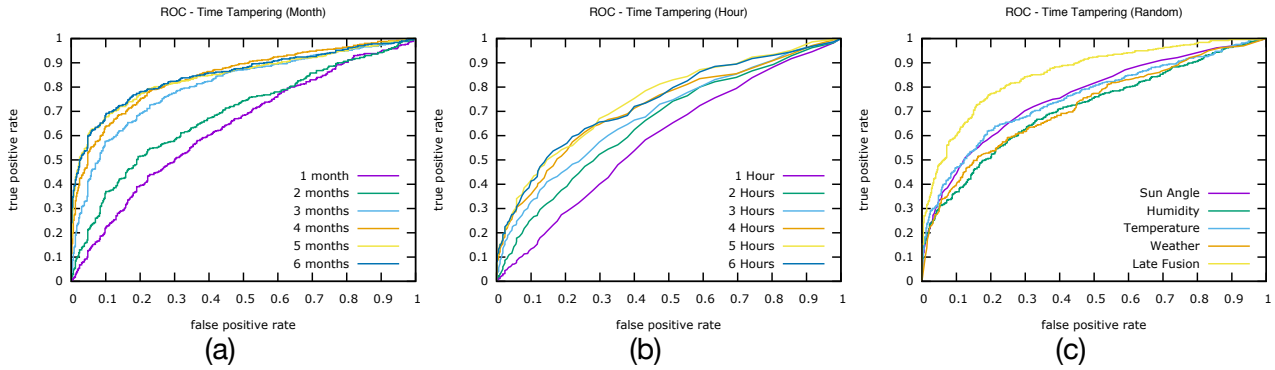

Figure 10. (a) ROC curves for the temperature model based monthly time tampered data detector. The different plots are for different variance noise in months added to tamper with metadata. The best performance expected is at a variation of 6 months, when the maximum seasonal variation is observed. (b) ROC curves for the sun altitude angle based hourly time tamper detector. The different plots are for different variance noise in hours that was used to modify the image time meta-data. The maximum sun altitude angle variation should be when the time difference is about 6 hours, which is what we can observe here. (c) ROC curve for time tampered data with timestamp changes ranging from one hour to one year. By combining all four models with late fusion, we can achieve better performance for tamper detection with an AUC of 85.5%.

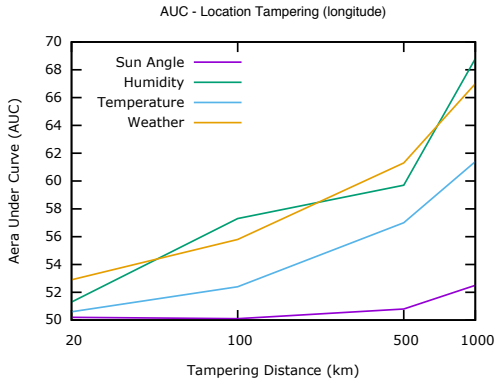

Figure 11. AUC on longitude tampered data. The performances of all models increase as the tampered distance increases.

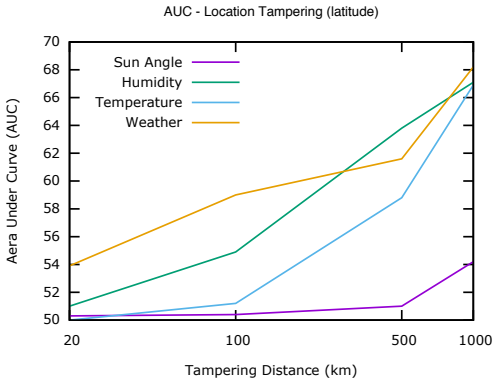

Figure 12. AUC on latitude tampered data. Compared to Figure 11, temperature model has better performance because temperature changes more drastically along different latitudes.

longitude tampered test set with different models. The performance of each model increases as the tampered distance increases because of a larger change in meteorological features. However, the sun angle model does not perform well in this case, because 1000km is too short a distance to have any detectable sun angle variation.

Figure 12 shows the AUC of tampering detection on latitude tampered test set. Temperature model has better performance on this test set compared to the previous one because temperature changes are more noticeable in different latitudes.

7. Conclusion

We propose a joint multi-task learning model to predict meteorological information from an image and use it to detect image metadata tampering. Our experiments show that joint multi-task model achieves better performance compared to any one model, and using the joint model we can detect different types of image metadata tampering with reasonable accuracy. Currently, we only apply simple late fusion to combine models for different meteorological information for tampering detection. Different ways to combine the models for meteorological information can be exploited in the future to further improve the detection results.

Acknowledgment This work was supported by the DARPA MediFor program under cooperative agreement FA87501620191, "Physical and Semantic Integrity Measures for Media Forensics."

References

[1] http://abcnews.go.com/wnt/story?id=1848351&page=1.

[2] Pysolar: http://pysolar.org/.

[3] Weather underground: www.wunderground.com.

[4] W.-T. Chu, X.-Y. Zheng, and D.-S. Ding. Image2weather: A large-scale image dataset for weather property estimation. In *Multimedia Big Data (BigMM), 2016 IEEE Second International Conference on*, pages 137–144. IEEE, 2016.

[5] D. Glasner, P. Fua, T. Zickler, and L. Zelnik-Manor. Hot or not: Exploring correlations between appearance and temperature. In *Computer Vision (ICCV), 2015 IEEE International Conference on*. IEEE, 2015.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[7] M. Islam, N. Jacobs, H. Wu, and R. Souvenir. Images+ weather: Collection, validation, and refinement. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop on Ground Truth*, volume 6, page 2, 2013.

[8] N. Jacobs, N. Roman, and R. Pless. Consistent temporal variations in many outdoor scenes. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–6. IEEE, 2007.

[9] P. Kakar and N. Sudha. Verifying temporal data in geotagged images via sun azimuth estimation. *IEEE Transactions on Information Forensics and Security*, 7(3):1029–1039, 2012.

[10] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Estimating natural illumination from a single outdoor image. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 183–190. IEEE, 2009.

[13] C. Lu, D. Lin, J. Jia, and C.-K. Tang. Two-class weather classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3718–3725, 2014.

[14] H. T. Sencar and N. Memon. Overview of state-of-the-art in digital image forensics. *Algorithms, Architectures and Information Systems Security*, 3:325–348, 2008.

[15] A. Volokitin, R. Timofte, and L. Van Gool. Deep features or not: Temperature and time prediction in outdoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 63–71, 2016.

[16] Z. Zhang and H. Ma. Multi-class weather classification on single images. In *Image Processing (ICIP), 2015 IEEE International Conference on*, pages 4396–4400. IEEE, 2015.