Scientific Workflows in Data Analysis: Bridging Expertise Across Multiple Domains

Ricky J. Sethi

Fitchburg State University

Yolanda Gil

USC Information Sciences Institute

Abstract

In this paper, we demonstrate the use of scientific workflows in bridging expertise across multiple domains by re-purposing workflow fragments in the areas of text analysis, image analysis, and analysis of activity in video. We highlight how the reuse of workflows allows scientists to link across disciplines and avail themselves of the benefits of inter-disciplinary research beyond their normal area of expertise. In addition, we present in-depth studies of various tasks, including tasks for text analysis, multimedia analysis involving both images and text, video activity analysis, and analysis of artistic style using deep learning. These tasks show how the re-use of workflow fragments can turn a pre-existing, rudimentary approach into an expert-grade analysis. We also examine how workflow fragments save time and effort while amalgamating expertise in multiple areas such as machine learning and computer vision.

1. Introduction

Scientific workflows help computational research in different disciplines by consolidating heterogenous software written in many different languages [1, 2, 3, 4, 5, 6]. Such workflows, designed by domain experts in their own fields, may also be of great utility to scientists in other disciplines; in fact, sites like `http://www.myexperiment.org` spotlight the opportunities for reusing and re-purposing scientific workflows [7, 8]. Although such sites help reproduce and reuse entire workflows, experts in other disciplines might want to use specific components of the workflows for new research purposes. Therefore, the ability to share components of workflows would allow researchers in different disciplines to compose applications that utilize the same functionality across very different domains of data.

An elegant solution to this research development problem is to utilize and share workflow fragments [6, 9, 10]. Workflow fragments are a coherent sub-workflow designed by a domain specialist. They have the potential to reduce

workflow authoring time and improve quality of the final workflows by allowing reuse of established, validated workflows. Each workflow fragment, in fact, is a useful resource in its own right and allows for cross-fertilization in new scientific domains [6].

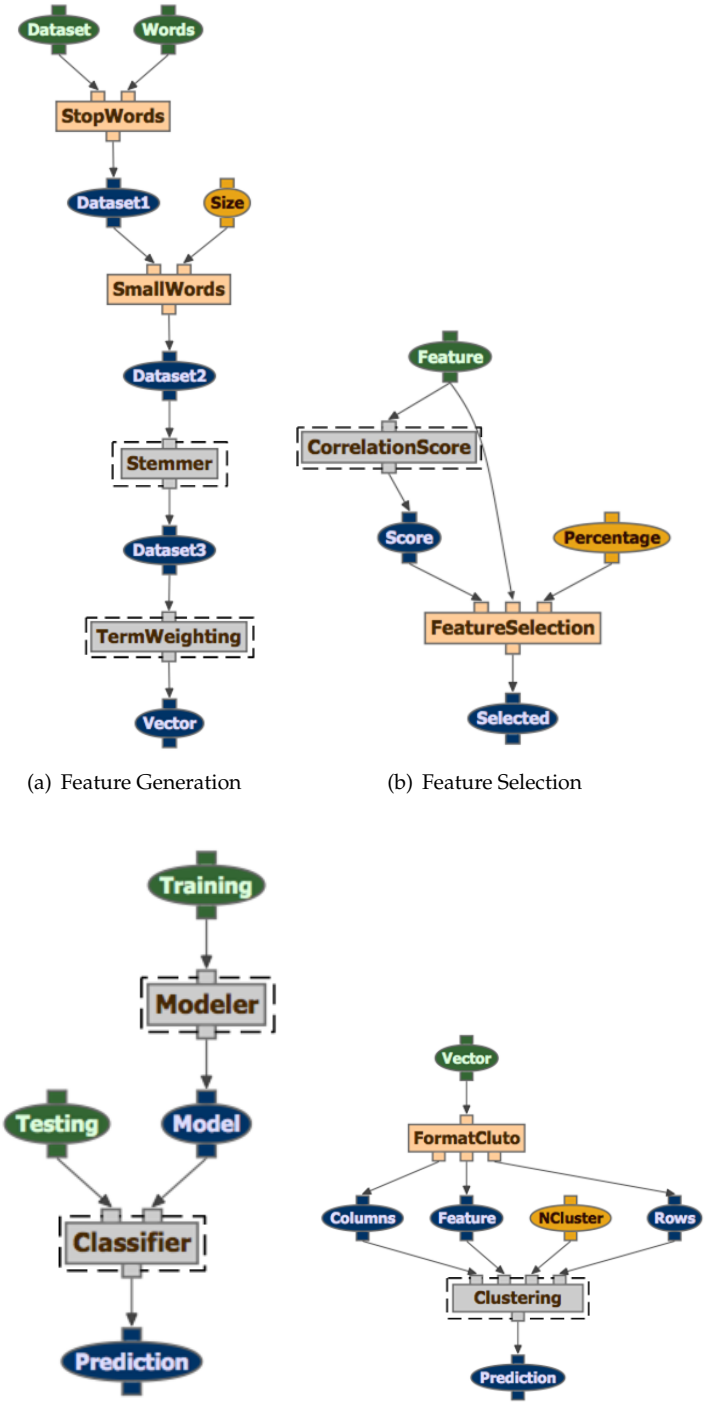
In this paper, we utilize workflow fragments to demonstrate the ability to reuse workflows as a way to facilitate development and bridge expertise across disciplines. We introduce several workflow fragments for text analysis, image analysis, analysis of activity in video, and analysis of artistic style using deep learning. In addition, we examine case studies that highlight the re-usability of workflow fragments across multiple data domains, from video analysis to multimedia analysis, which involves both text and image analysis. In particular, we show how a pre-existing but incomplete multimedia analysis task can be developed more rapidly and extended by simply reusing workflow fragments we had previously developed. This new workflow can subsequently be made accessible to end-users or researchers to conduct further analysis or reproduce results, as needed.

To facilitate this export of workflows and workflow fragments, we utilize the WINGS workflow system, which was developed to assist scientists in managing complex computations [3] and has been used in several scientific applications [5]. WINGS uses semantic workflow representations that capture the requirements and semantic constraints of individual steps and datasets explicitly, as well as workflow reasoning algorithms to generate and validate possible combinations of workflow components systematically.

Our main contributions in this paper are:

- Creation of various workflow fragments for text analysis, image analysis, analysis of activity in video, and analysis of artistic style using deep learning.
- Case studies that show the re-usability of workflow fragments across multiple data domains, including computer vision and machine learning applications for multimedia analysis.
- Analysis of development time and effort to both extend a nascent, rudimentary analysis of a multimedia analysis project using the provided workflow fragments and to port its pre-existing code as new workflow fragments as well as creating multiple implementations of the neural algorithm for artistic style.

The rest of this article is organized as follows. In Section 3, we discuss the WINGS workflow system. In Section 4, we show several workflow fragments we created for text analysis, image analysis, analysis of activity in video, and analysis of artistic style which incorporate vastly hetergenous codebases. In Section 5, we provide an in-depth case study of the re-use of workflow fragments to extend a rudimentary multimedia analysis task. We then demonstrate how the workflow fragments we created can be re-used to enable rapid development and deployment of several research projects in Section 7. Finally, in



(a) Feature Generation (b) Feature Selection

(c) Classification. (d) Clustering.

Figure 1: Workflow Fragments developed for Text Analysis [11]. Here we see workflow fragments for a) Feature Generation; b) Feature Selection; c) Training and Classification; and d) Clustering.

Section 9, we highlight how the re-use of workflows allows savings of time and effort as we link across disciplines, followed by a discussion of future directions in Section 10.

2. Related Work

Several workflow systems are used for scientific applications, including Pegasus [12], Taverna [13], Vistrails [14], and Kepler [15]. Other scientific workflow engines are used in specific domains, such as GenePattern [16] and Galaxy [17] for omics and LONI Pipeline for neuroimaging. These tools enable users to create reusable workflows to support user-defined tasks. Although all these workflow engines offer very useful capabilities, they focus on capturing low-level mechanics of the software to run at each step. But data analysis also involves higher-level processes that require reasoning, such as whether an algorithm will work with a new dataset or how to set parameters to get the best results. The expert knowledge necessary to make such decisions is not available in current workflow systems. WINGS [18] is unique in automating the generation of workflows by selecting algorithms, data, and parameter settings.

Our approach uses the WINGS workflow system, which has three key features that make workflows accessible to end-users: a simple dataflow structure, an easy-to-use web interface [19], and an ability to export workflows and workflow fragments as web objects [20]. This framework allows us to structure computer vision and machine learning methods as computational workflow fragments described in high-level declarative notations and capable of processing large quantities of data that comes from multiple sources or files [3, 21]. WINGS is open source, built upon open web standards from the World Wide Web Consortium (W3C), and is available at `http://www.wings-workflows.org/`.

3. Making Workflows Accessible to End-Users: Semantic Workflows and WINGS

Using a semantic workflow system like WINGS to assist with the design of computational experiments allows for creating structured, systematic experiments that can be automated, thus allowing anyone to re-purpose either entire workflows or just workflow fragments. In addition, the WINGS workflow system has an open modular design and can be easily integrated with other existing workflow systems and execution frameworks to extend them with semantic reasoning capabilities.

The WINGS workflow system is pre-equipped with several expert-quality workflows that represent a powerful set of analytic methods [22, 23]. It includes workflow fragments for general machine learning packages like Weka [24], document clustering packages like CLUTO, topic modeling algorithms like LDA, etc. We extend these repositories by creating workflow fragments based on popular computer vision and machine learning packages like OpenCV

[25], a standard computer vision library, and MALLET [26], a standard package for statistical processing and information extraction, as well as adding custom implementations of some state-of-the-art computer vision/machine learning models, including convolutional neural networks.

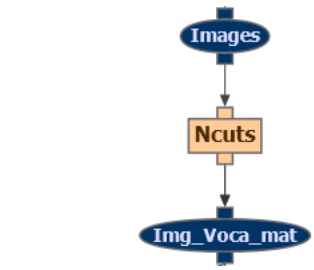
These packages have vastly heterogeneous implementations but the workflow fragments encapsulate the software with interfaces described by data types in the workflow system to make them reusable in different workflows. WINGS ensures that only the right components are used in workflows by checking the semantic constraints of the input and output types for every component. The system ensures that only workflows with valid combinations of components are executed. The framework also includes several widely used datasets used for comparison purposes in the text analytics and computer vision community.

In addition, these workflow fragments can be exported in WINGS by publishing them as web objects using Linked Data principles [20] and can be made available as part of a workflow library. These web objects, represented in RDF, allow direct access via unique URIs to workflow fragments or workflows, their components, and their associated datasets. Such web objects can then be imported into any workflow system that is compatible with the PROV or Open Provenance Model standards for workflow publication [27, 28] so that other researchers can directly re-use or re-purpose any single workflow fragment or entire workflows.

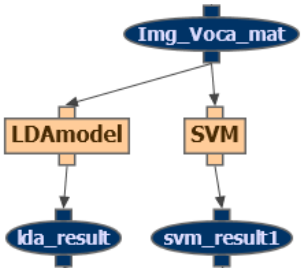
4. Workflow Fragments for Text Analysis, Image Analysis, and Analysis of Activity in Video

Workflow fragments represent multi-step methods that can be easily reused across workflows. Such predefined workflow fragments make complex analytics expertise readily available to new users. The components that make up workflow fragments can be written in heterogeneous languages: e.g., some components are in Java, others in Matlab, and still others in C++ but the language of choice is irrelevant as the components are integrated into the workflows without reliance upon their individual implementation idiosyncrasies. This is possible because each individual program is converted into a workflow component via a short wrapper shell script (usually 3-5 lines of code) thus allowing any pre-existing program to be incorporated as a new component in a workflow or workflow fragment.

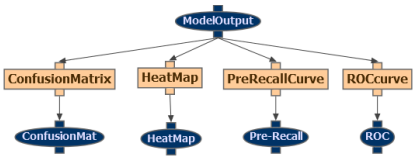
These previously defined workflow fragments can be executed independently from each other. This is helpful as some researchers might choose to focus on particular parts in order to optimize or improve their understanding of the behaviour in the individual steps. A good starting point to build a new application is to create end-to-end workflows that are formed by re-using and re-purposng workflow fragments. These end-to-end workflows would then incorporate and represent advanced expertise in that they would capture complex combinations of components that are known to work well in practice. Such re-usable workflow fragments are pre-defined by domain experts and



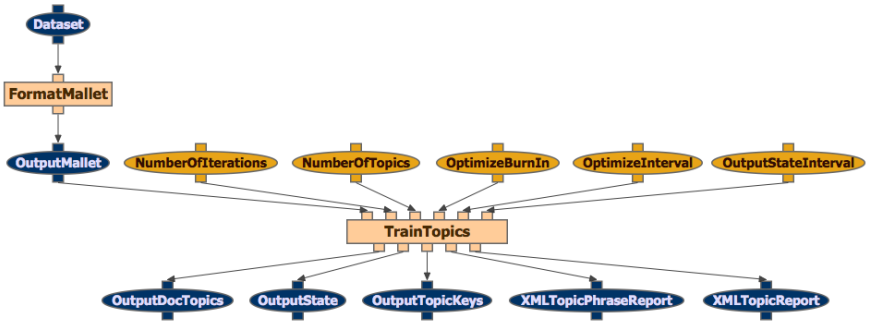
(a) N-Cuts Image Segmentation



(b) LDA and libSVM

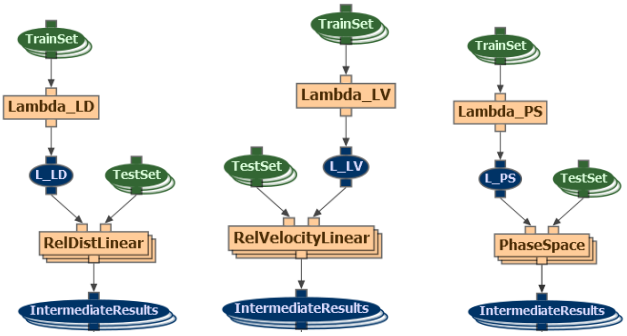


(c) Statistics Evaluation

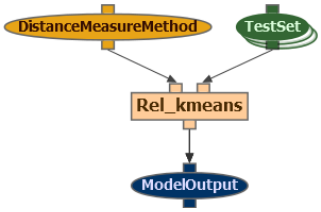


(d) Topic Models

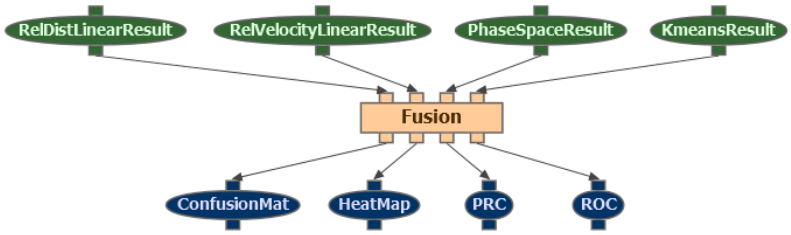
Figure 2: Workflow Fragments for Image Analysis. Here we see workflow fragments for a) N-Cuts Image Segmentation; b) Latent Dirichlet Allocation and Support Vector Machines (libSVM); c) Statistics Evaluation (Confusion Matrices, Heatmaps, Precision Recall Curves (PRC), and Receiver Operating Characteristic (ROC) Curves); and d) Topic Modelling (Mallet).



(a) Relative Distance (b) Relative Velocity (c) Phase Space



(d) K-Means



(e) Fusion: weighted average of all other methods

Figure 3: Workflow Fragments for five Video Activity Recognition Models. Here we see workflow fragments for a) Relative Distance with Linear Fit; b) Relative Velocity with Linear Fit; c) Phase Space as Relative Distance with Exponential Fit; d) Relative Distance with k-means clustering; and e) Fusion: a weighted average of all other methods.

available as part of workflow libraries. They can be executed with available datasets or adapted by adding or changing components.

4.1. Workflow Fragments for Text Analysis

Here, we detail some of the workflow fragments we have developed [11] for Text Analysis as seen in Figure 1.

4.1.1. Text Pre-Processing and Feature Generation

Analytic tasks usually begin with some preprocessing steps to generate the features of a document. The workflow fragment for feature generation is shown in Figure 1(a). Morphological variations are removed from the dataset with a stemmer component. The WINGS workflow system provides several choices, including a Porter Stemmer and a Lovins Stemmer. It further provides term weighting components that are used to transform the dataset into the vector space model format. Among them are term frequency-inverse document frequency, corpus frequency or document frequency for instance. The generated outcome can now be used with different other workflows and is independent of a particular implementation at this stage in the workflows.

4.1.2. Feature Selection

A very common step for many classification problems is feature selection, as shown in Figure 1(b), whose main purpose is to reduce the training set by only using the most valuable features. This will reduce the necessary time for training the model and can improve the results of the classifier in some cases. The goodness of a feature in the dataset is measured with the correlation score. Typical implementations for this step are Chi Squared, Mutual Information or Information Gain that can be found in [29] and are all implemented in the framework. The resulting score is used in a feature selection step to retain the most valuable features in the training set. The percentage of selected features is typically changing for every dataset respectively classifier used in the computational experiment.

Another characteristic for this workflow fragment is that it uses heterogeneous implementations for the components. While the components for the computation of the correlation score take advantage of the capabilities of MATLAB to handle large matrices very elegantly, the component for the feature selection uses an implementation written in Java.

4.1.3. Classification and Clustering

The resulting training set after the feature selection can be used for the training of a model with the workflow fragment shown in in Figure 1(c). Both components in the workflow use the Weka machine learning framework. Thus, many different machine learning algorithms can be used to perform experiments with the dataset. Among them are very popular algorithms from the text analytic community like Support Vector Machines, Naive Bayes or k-Nearest Neighbor. The computed model can be stored in the data catalog and reused

for later classifications. Since the training is usually a very time demanding task in the workflows, it is very desirable to reuse previously created models. Existing models are also easier to compare against each other, because the metadata information of the model carries provenance information from the components used and their configuration during the workflow execution. In the second step a classifier uses the trained model with the testing set to compute the predictions.

In Figure 1(d), we see the workflow fragment for clustering. The Vector that results from the Feature Generation workflow fragment in Figure 1(d) can be used as input for clustering. It needs to be formatted into the suitable format for the clustering software. The result of this step is the Feature output with the transformed Vector. Next to this output there are additional intermediate files called Rows and Columns that contain the label names that are used to annotate the final result with the right names for the features and labels. The parameter for this component is used to specify the number of clusters to be applied on the data set.

4.2. Workflow Fragments for Image Analysis

Here, we detail some of the Workflow Fragments we have developed for Image Analysis as seen in Figure 2. In particular, we created workflow fragments for a) Normalized Cuts Image Segmentation [30] which views image segmentation as the optimal partitioning of a graph by minimizing the cut with a modified cost function; b) Latent Dirichlet Allocation (MALLET) [26] for visual-word clustering and Support Vector Machines (libSVM) for visual-word classification [31]; c) Statistics Evaluation (confusion matrices, heatmaps, Precision Recall Curves (PRC), and Receiver Operating Characteristic (ROC) Curves) [32, 33, 34]; and d) Topic Modelling (MALLET) [26] for video-word clustering. In particular, the statistics evaluation workflow fragment allows for easy visualization of diverse summary and graphical statistical measures which are the outputs of that component (i.e., summary measures like equal error rate, mean average precision, etc., as well as the graphical outputs of confusion matrices, heatmaps, precision recall curves, and receiver operating characteristic curves). Visual-words and video-words are the image and video equivalent of text words used in textual bag-of-words models; in computer vision, they are created by partitioning an image or video into interest point cuboids or segments and then computing some features (for which it is possible to calculate a distance metric) for each interest point cuboid. The centers of each of these clusters are the visual words (codewords) in the visual vocabulary (codebook).

4.3. Workflow Fragments for Analysis of Activity in Video

Here, we detail some of the workflow fragments we have developed for video activity recognition as seen in Figure 3. We implemented the most appropriate models for the ISI Atomic Actions datasets [35, 36] (which consists of 190 videos as explained in Section 7), derived from [37, 38, 39, 40, 41, 42, 43, 44],

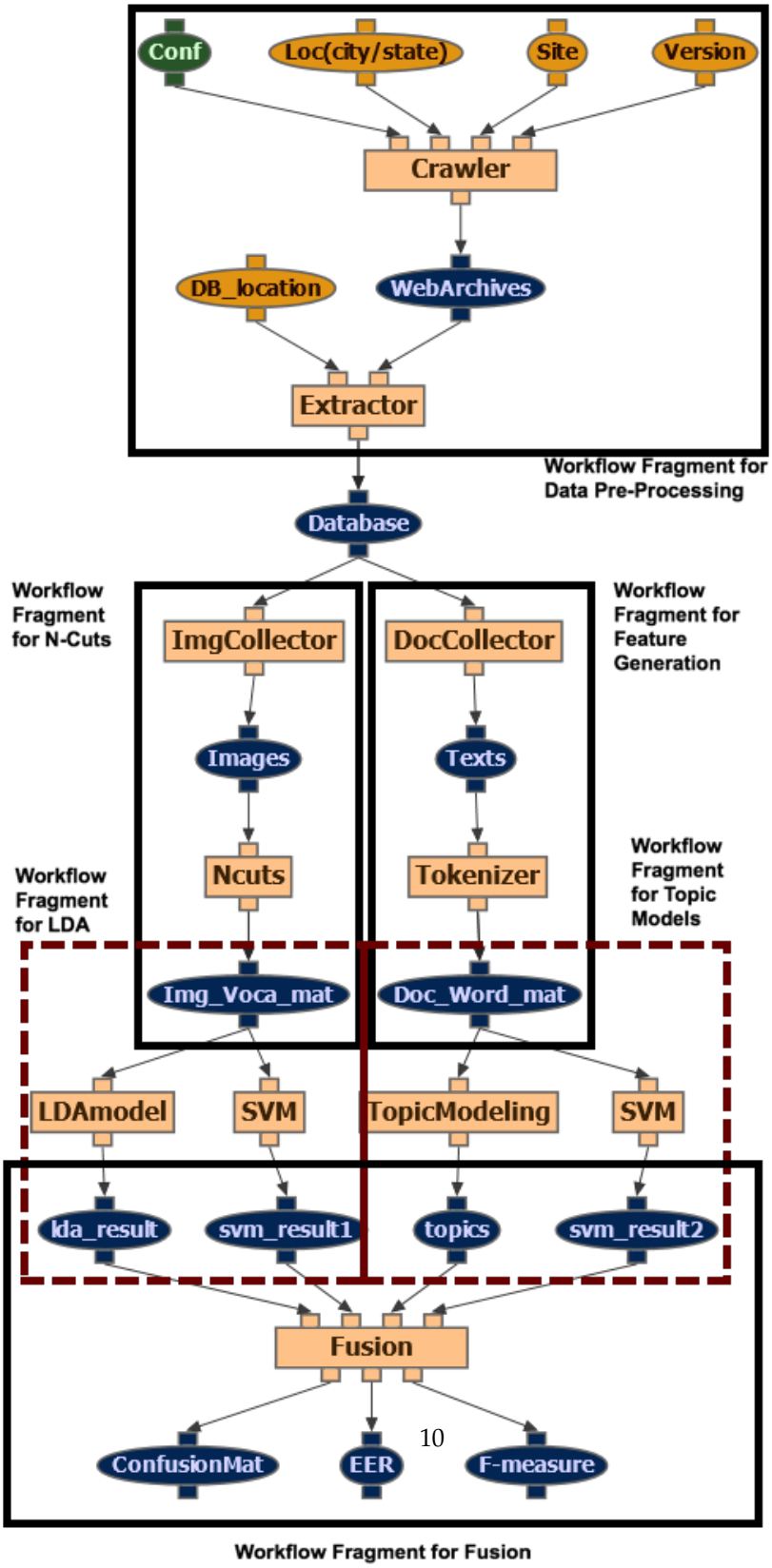


Figure 4: Fusion of Image and Text Analyses in the MultiMedia Analysis workflow.

as workflow fragments in the WINGS workflows system; the workflow fragments corresponding to these five models for video activity recognition are shown in detail in Figure 3 and are: a) Relative Distance with Linear Fit; b) Relative Velocity with Linear Fit; c) Phase Space as Relative Distance with Exponential Fit; d) Relative Distance with k-means clustering; and e) Fusion: a weighted average of all other methods, which outputs the graphical statistical measures confusion matrices, heatmaps, precision-recall curves, and receiver operating characteristic curves, as well as summary statistical measures of equal error rate, mean average precision, etc.

5. Case Study: MultiMedia Analysis

We demonstrate the utility of the image analysis and text analytics workflow fragments by extending a pre-existing text analysis for a multimedia analysis task that tries to detect human trafficking. This project analyzes posts on various sites on the internet in order to determine whether the subject of that post is a victim of human trafficking. The ultimate goal of this project is to create intelligence which may be used by law enforcement to detect and combat trafficking by making a determination of whether or not the subject of a post was trafficked or not.

The initial development of the project had progressed to creating a crawler, which downloads posts from various posting sites, and an extractor, which extracts the text and images and stores them in a database. However, there had been no substantial analysis of the posts in this nascent project. We extended the project to examine both the text of the post (using the text analytics workflow fragments we had already developed), as well as the associated images (using the image analysis workflow fragments we developed); a final determination about trafficking of the subject of the post was made by fusing the results of the Text and Image analysis via the fusion workflow fragment we developed. The goal of this project is to use both the text and image content of posts to make a stronger determination of whether or not the subject of the post was trafficked. Thus, the re-use and re-purposing of workflow fragments allowed a multimedia analysis spanning data domains of text and image analysis, including the fusion of their results in the final determination.

In particular, we componentized, re-used, and re-purposed the following workflow fragments in the multimedia analysis task:

- Componentized: Crawler and Extractor Workflow Fragment
- Re-used: N-Cuts, Feature Generation, LDA, and SVM Workflow Fragments from Figures 1 and 2.
- Re-purposed: Fusion Workflow Fragment from Figure 2.

We first componentized the previously developed crawler and extractor as workflow fragments using the WINGS framework and then re-used/re-purposed

our workflow fragments to create the final workflow for human trafficking detection. The resulting workflow is shown in Figure 4 where the top black box labelled "Componentized Workflow Fragment" shows the original crawler and extractor incorporated as components in WINGS. This is followed by:

- Re-Use: The next two boxes labelled "Workflow Fragment for N-Cuts" and "Workflow Fragment for Feature Generation" show re-use of the Image Analysis workflow fragments from Figure 2 as well as the re-use of the Text Analytics workflow fragments from Figure 1, respectively. Here, the "Tokenizer" component represents the entire workflow fragment in Figure 1(a).
- Re-Use: The next two boxes labelled "Workflow Fragment for LDA" and "Workflow Fragment for Topic Models" show the re-use of workflow fragments for unsupervised analysis using MALLET and supervised analysis using SVM in a bag-of-words model from both the Image Analysis workflow fragments in Figure 2 and the Text Analytics workflow fragments in Figure 1. Here, the "TopicModeling" component represents the entire workflow fragment in Figure 2(d).
- Re-Purpose: The final box labelled "Workflow Fragment for Fusion" shows the re-purposed Fusion workflow fragment from Figure 3 for fusing the results of the Text and Image Analysis and visualizing those results.

Summary results from the Fusion module showed an Equal Error Rate of 0.37 and F-Measure of 0.47.

6. Case Study: Text Analysis

We demonstrate the utility of the Image Analysis workflow fragments we created by re-using and re-purposing them to enable rapid development and deployment of several research projects spanning diverse data domains, including the analysis of the text questions and answers on the The Madsci Network.

The Madsci Network is an Ask-A-Scientist website [45] . It provides a human-mediated Question & Answering (Q&A) service that answers questions in 26 different scientific fields. Boasting a store of over 40,000 questions and answers, it serves as a unique repository of scientific knowledge. However, with more than 650,000 unique visitors and only 700 scientists to answer questions, it is worth automating some of the processes that are currently done manually to handle user questions.

When a user first comes upon the site, they might immediately try to submit a question they have wrestled with for a while, as shown in the question process flow in Figure 6. One of the issues with the operation of such an immense knowledge base is that it is difficult to automatically determine whether a new question has already been answered on the website. If it has not, the question is routed by the moderator to a scientist who is best suited to answer that

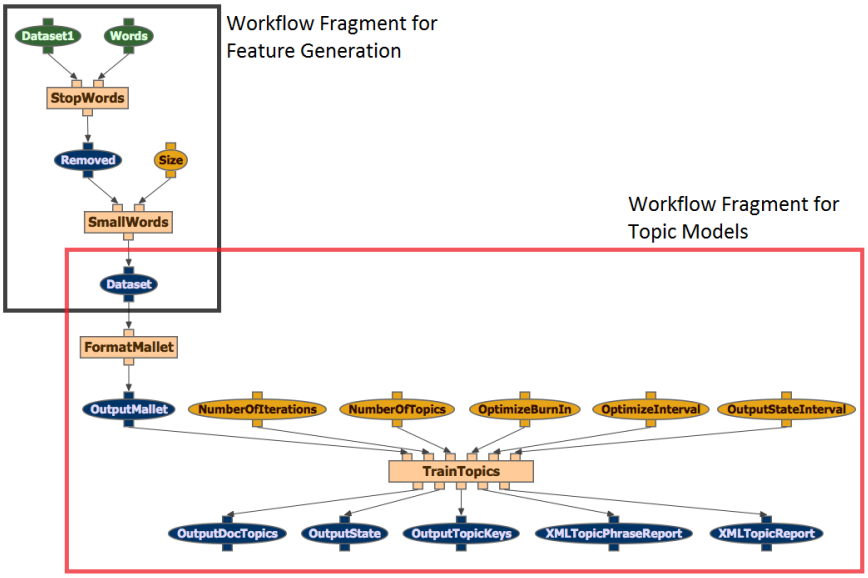


Figure 5: Topic Modeling Workflow.

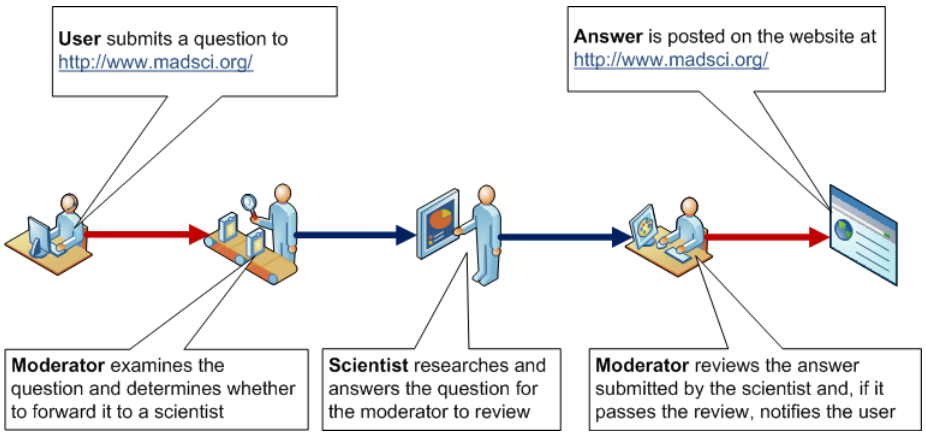


Figure 6: Overview of the question processing flow on The MadSci Network.

query. However, finding scientists that are especially appropriate for a specific question is equally challenging given the vast number of scientists actively answering questions on the site. Finally, determining the correct category into which a question falls is another substantial machine learning task associated with Q&A sites as users often mis-categorize their queries. Thus, the main questions associated with analysis of The Madsci Network corpus are:

1. Automatic Question Answering: suggesting best matches from the archives for an incoming question
2. Task Assignment/Expert Finding: finding the best-suited scientist for incoming questions
3. Label Assignment: finding the most appropriate category for incoming questions

Concomitant with these research thrusts are several other issues, including dealing with short documents (e.g., the lengths of submitted questions,) and examining trends in the data that have applicability well beyond the specific corpus studied. A promising new approach to help address all of these data analysis problems is based on topic modeling. Topic models are a Bayesian graphical model-based approach to discovering hidden semantic topics in a corpus. One of the most popular tools which implements Latent Dirichlet Allocation, and its many variations, is MALLET, which is used in the WINGS topic modeling workflow. Just as with other machine learning methodologies applied to a specific corpus, topic models require in-depth and varied experimentation. Once the theoretical models have been established, significant experimentation is needed to determine model selection and parameter optimization, output analysis, and extensive evaluation of results for various experimental scenarios. This is especially important in topic modeling as no formal, structured approach to evaluation currently exists. Once the initial analysis and baseline is established, new models can be implemented and compared to the baselines.

The **Madsci Topic Modelling Workflow**, shown in Figure 5, re-uses the Feature Generation Workflow Fragment from Figure 1 and the Topic Model Workflow Fragment from Figure 2. The various parameters associated with MALLET, the topic modeling framework utilized here, as well as the various outputs, can all be easily specified, customized, and used in subsequent processing.

We used the WINGS workflow for topic modeling shown in Figure 2. The various parameters associated with MALLET, as well as the various outputs, can all be easily specified, customized, and used in subsequent processing, as shown below. For example, we can easily take one of the MALLET outputs, the OutputDocTopics, which shows the distributions over topics for each document, and insert a Weka component to visualize it. This visualization is shown in Figure 7. This is the plot of a single question, and its distribution over topics, which clearly shows the dominance of a single topic in the distribution. Such plots intuitively reveal insights about the individual questions and about the

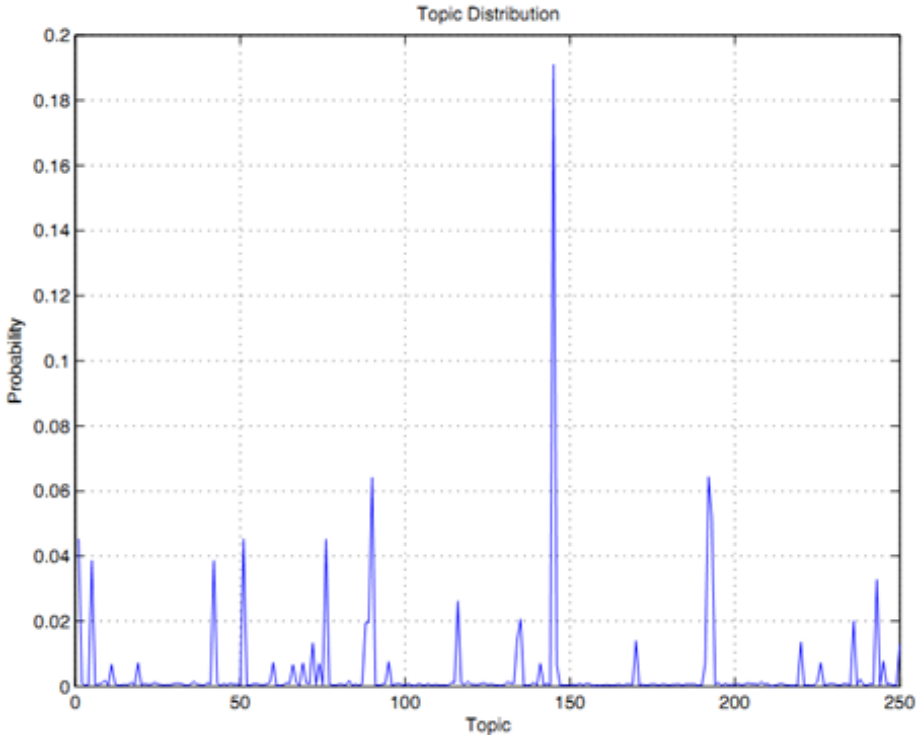


Figure 7: Topic distribution for a sample question on The Madsci Network.

overall dataset. In addition, this kind of visualization would easily allow comparison of the histograms of similar questions in order to determine the most similar questions and answers using simple distance measures which are inserted as components in the additional processing of the MALLET output. Initially, we got results for that experiment that had many category labels. Later, we used coarser-grained category labels for each document where the coarser grain categories are super-sets of the original labels. Table 1 shows the confusion matrix for the new categories. The initial results as well as examples of workflow variations created can be found in the project website.

It is also relatively simple to analyze how questions and answers cluster together, using the clustering workflow fragments from Figure 1. The results are shown in Figure 8. We can use this workflow to show how documents and topics cluster; this can be used by both the users and the moderator. When a new question is submitted, a new clustering diagram will be produced in which topics would be on the y-axis and documents on the x-axis; this would clearly show which questions/answers cluster with the correct answer (on the x-axis) for the user and which topics cluster together on the y-axis for the moderator to see which topics are most relevant for the new question. The main realization was that using the WINGS workflow system simplified the process of analysis

	a	b	c	d	e	f	g
a	49	4	19	1	3	34	2
b	3	46	28	3	3	42	2
c	7	23	390	34	7	41	5
d	8	6	56	93	7	19	3
e	2	5	11	0	9	9	2
f	27	21	42	11	13	478	3
g	4	4	9	2	1	8	5

Table 1: Confusion matrix for coarse-grained categories.

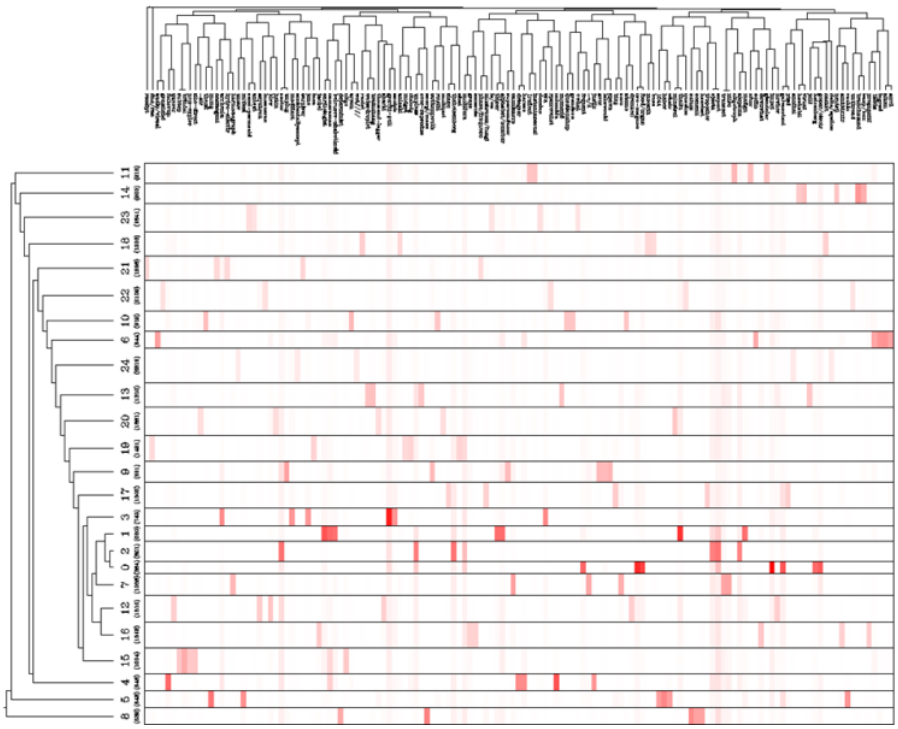


Figure 8: Clustering output for The Madsci Network dataset.

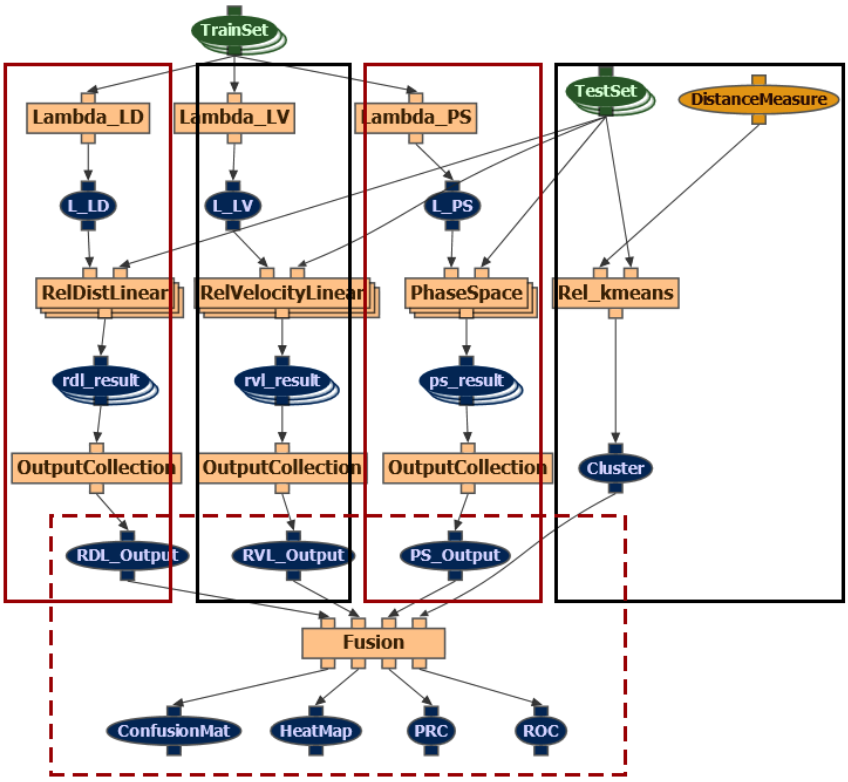


Figure 9: Overall Workflow for the Atomic Pair Actions Dataset Analysis. Some of the individual component workflow fragments, like the Video Activity Recognition models and Fusion module, that are re-used in this overall workflow are shown in Figure 3.

significantly. It not only allowed calculation of standard statistics but also facilitated plotting of document-topics to help visualize it using CLUTO, allowed extensions of the MALLET toolkit (e.g., the Poly-Lingual Topic Model, PLTM, as well as new custom models) to be incorporated easily, with just as trivial replication of previous experimentation, allowed post-processing and visualizing of complex text output as shown in the Precision-Recall and ROC curves, as well as the histogram spectra of topic distributions, using tools like Weka.

7. Case Study: Analysis of Activities in Video

In this section, we further demonstrate how the workflow fragments we created can be re-used and re-purposed to video activity recognition in the Atomic Actions dataset analysis.

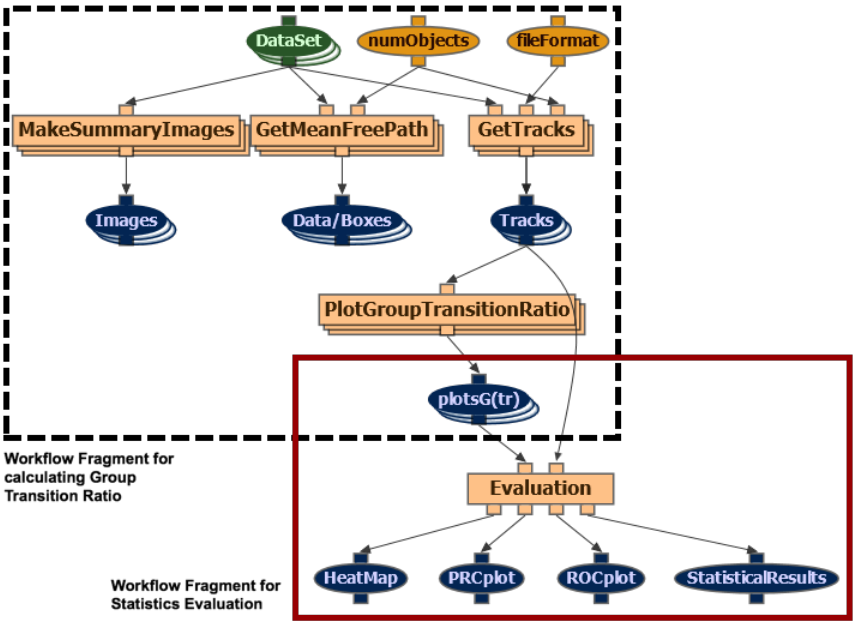


Figure 10: Workflow for calculating the Group Transition Ratio (G_{tr}) model, used for the Atomic Group Actions Dataset, which re-purposes the Statistics Evaluation workflow fragment from Figure 2.

The **Atomic Pair Actions Workflow** re-uses the Video Activity Recognition Workflow Fragments and Statistics Workflow Fragments from Figures 2 and 3. Similarly, the **Atomic Group Actions Workflow** re-purposes the Statistics Modules from Figure 2, in addition to implementing custom workflow fragments for the Group Transition Ratio (G_{tr}) model [40, 39, 36].

The ISI Atomic Pair and Group Actions Dataset contains videos obtained from a combination of YouTube or public domain datasets. It examines Atomic Pair Actions (converging, parallel, diverging) as well as Atomic Group Action (group formation, dispersal, and movement) [40, 39, 36, 35, 46]. The entire dataset contains 190 videos and some sample frames are shown in Figure 11.

In Figure 9, we see the overall workflow for the Atomic Pair Actions Dataset analysis. Some of the individual component workflow fragments, like the Video Activity Recognition models and Fusion module, that are re-used in this overall workflow are shown in Figure 3. Then, in Figure 10, we see the workflow for calculating the Group Transition Ratio (G_{tr}) model, used for the Atomic Group Actions Dataset, which re-purposes the Statistics Evaluation workflow fragment from Figure 2.

In both of these workflows, we re-use or re-purpose the Statistics Evaluation workflow fragment from Figure 2 to calculate heatmaps, PRC curves, ROC

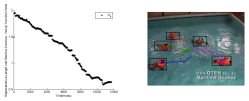
Figure 11: Sample frames from video clips from the ISI Atomic Actions datasets.

curves, confusion matrices, etc., as shown in Figure 14, in addition to various summary statistical measures like Equal Error Rate, F-Measure, Mean Average Precision, etc., as shown in Table 2.

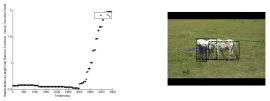
In Figure (12) (a) - (c), we show the G_{tr} vs. time graph and a sample frame for all three group-group Atomic Group Actions: *group formation*, *group dispersal*, and *group movement*. They show how the G_{tr} decreases with time for *group formation* in (a), increases with time for *group dispersal* in (b), and stays the same for *group movement* in (c). In Figure (12) (d) - (e), the G_{tr} vs. time graph gives an initial categorization but then we also need to use the graph of trajectories to differentiate between the group-person Atomic Group Actions of *person joining* in (d) and *person leaving* in (e).

We also show some quantitative results using standard statistical evaluation. In Figure 13, we show the ROC curve, PRC, and Heatmap for the group-group actions: Category 1 (*group formation*), Category 2 (*group dispersal*), and Category 3 (*group movement*). They show that Category 3 was hardest to classify while Category 1 was the easiest. Please note: Category 3 has the smaller square in the heatmap as group-person action videos were added to Category 1 and 2, doubling their size. We also show the Confusion Matrices, EER, and F-Measure for both group-group and group-person atomic group actions in Table 2. For the overall table of confusion for *all* atomic group actions, we had TP=0.12, FN=0.08, FP=0.08, and TN=0.72. Our main purpose in this analysis is to quantify group-group atomic group actions using the G_{tr} and we use a simple baseline method based on trajectories for group-person classifications which results in lower classifications. Other issues affecting accuracy include videos with small number of objects for clustering (three or less lead to lower results), sufficient number of videos used for training (dependent on the method utilized), and perspective issues with motion in the z-direction (to be improved in future tracker implementations).

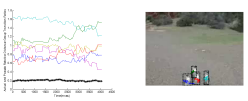
Here we see a variety of different levels of workflow fragment componentization, re-use, and re-purposing: in Figure 4, we componentize two components, re-use four different workflow fragments, and re-purpose one workflow fragment, all analyzing *image* and *text* data; finally, in Figure 9, we directly re-use five workflow fragments and, in Figure 10, we componentized four new components and re-purposed one workflow fragment, all for the analysis of *video* data.



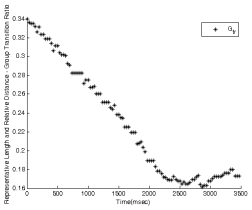
(a) *Group Formation*: G_{tr} decreases from about 2 to about 0.5. Sample image shows swimmers converging upon a central object, forming a group.

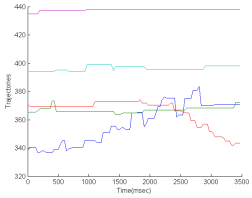


(b) *Group Dispersal*: G_{tr} increases from about 0.5 to about 2.0. Sample image shows a group of horses dispersing when startled.

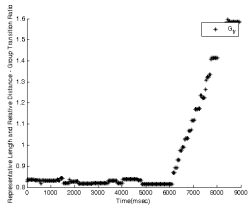


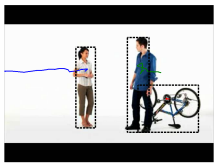
(c) *Group Movement*: G_{tr} stays steady at about 0.2. Sample image shows penguins waddling along together.

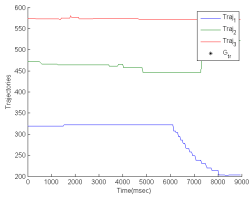




(d) *Person Joining*: G_{tr} decreases from about 0.8 to about 0.6. Sample image shows a baseball player joining other baseball players. For group-person action classification, the trajectories vs time graph (on the right) is used in addition to the G_{tr} vs time graph (on the left).

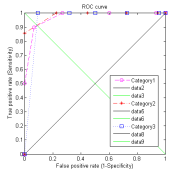


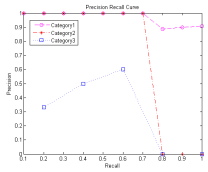




(e) *Person Leaving*: G_{tr} increases from about 0.9 to about 1.5. Sample image shows lady leaving group. For group-person action classification, the trajectories vs time graph (on the right) is used in addition to the G_{tr} vs time graph (on the left).

Figure 12: In (a) - (c), we show the G_{tr} vs. time graph and a sample frame for all three group-group Atomic Group Actions: *group formation*, *group dispersal*, and *group movement*. They show how the G_{tr} decreases with time for *group formation* in (a), increases with time for *group dispersal* in (b), and stays the same for *group movement* in (c). In (d) - (e), the G_{tr} vs. time graph gives an initial categorization but then we also need to use the graph of trajectories to differentiate between the group-person Atomic Group Actions of *person joining* in (d) and *person leaving* in (e).





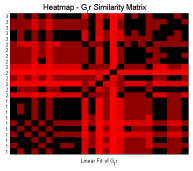
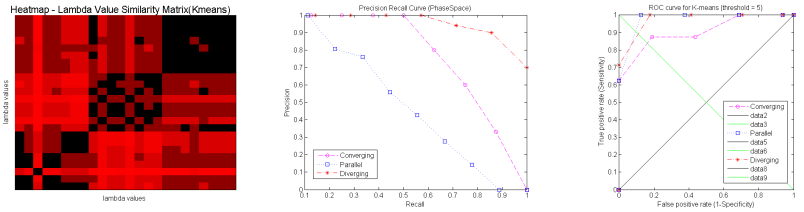


Figure 13: ROC curve, PRC, and Heatmap for the group-group actions: Category 1 (*group formation*), Category 2 (*group dispersal*), and Category 3 (*group movement*). They show that Category 3 was hardest to classify while Category 1 was the easiest. Please note: Category 3 has the smaller square in the heatmap as group-person action videos were added to Category 1 and 2, doubling their size.



(a) K-Means Heatmap (b) PS PRC curve (c) K-Means ROC Curve

Figure 14: Results showing Heatmaps, PRC, ROC curves, and Confusion Matrices of the best performing Video Activity Recognition models on the ISI Atomic Actions datasets.

	F-Measure			Equal Error Rate			Average Precision			MAP
	Converging	Diverging	Parallel	Converging	Diverging	Parallel	Converging	Diverging	Parallel	
RDL	0.928571	0.823529	0.814815	0.235300	0.176500	0.235300	0.875000	0.961700	0.642500	0.826400
RVL	0.235294	0.320000	0.800000	0.956500	0.250000	0.600000	0.000000	0.150900	0.672500	0.274500
PS	0.846154	0.736842	0.666667	0.277800	0.315800	0.176500	0.750000	0.957100	0.422000	0.709700
K-Means	1.000000	0.933333	0.941176	0.187500	0.125000	0.176500	0.930600	0.861100	1.000000	0.930600
FUSION	0.222222	0.777778	0.476190	0.000000	0.235000	0.467000	0.638900	0.640300	0.738500	0.672600

Table 2: Cumulative Statistical Comparison of all five video activity recognition models from Section 4.3 for the Atomic Pair Actions. Here we see the K-Means outperforms in the F-Measure, Equal Error Rate (EER), and Average Precision/Mean Average Precision (AP/MAP) for RDL (Relative Distance), RVL (Relative Velocity), PS (Phase Space), K-Means (Relative Distance with k-Means), and Fusion. RVL is the worst performer in all categories.

8. Case Study: Neural Algorithm Analysis of Artistic Style

In this section, we implement the Neural Algorithm for Artistic Style developed in [47]. In order to allow maximal flexiblity in the use of the Neural Algorithm of Artistic Style [47], we developed two separate implementations as shown in the two workflow fragments shown in Figure 15, which uses Lua and Torch, and Figure 16, which uses Python and Google's TensorFlow. Both of these use fragments for pre-processing that were also developed in Section 4.

The Neural Algorithm of Artistic Style uses deep learning (specifically, a Convolutional Neural Network, CNN) to separate the style and content of an image. It designates one image as a style image and one as target image. It then extracts the style from the style image and applies it to the content of the target image to create a new image in the style of the style image.

For example, Figure 17 shows the target image of a scene from Tubingen as presented in the original paper [47]. The algorithm then extracts the style from the Starry Night painting of Van Gogh. This style is applied to the Tubingen image to create a new image of Tubingen in the style of Van Gogh. Similarly, they extract the style of Munch using his painting, The Scream, and apply this to the Tubingen scene, as well. We reproduce these results in Figure 17 using the workflow fragments shown in Figure 15. We then applied those same frag-

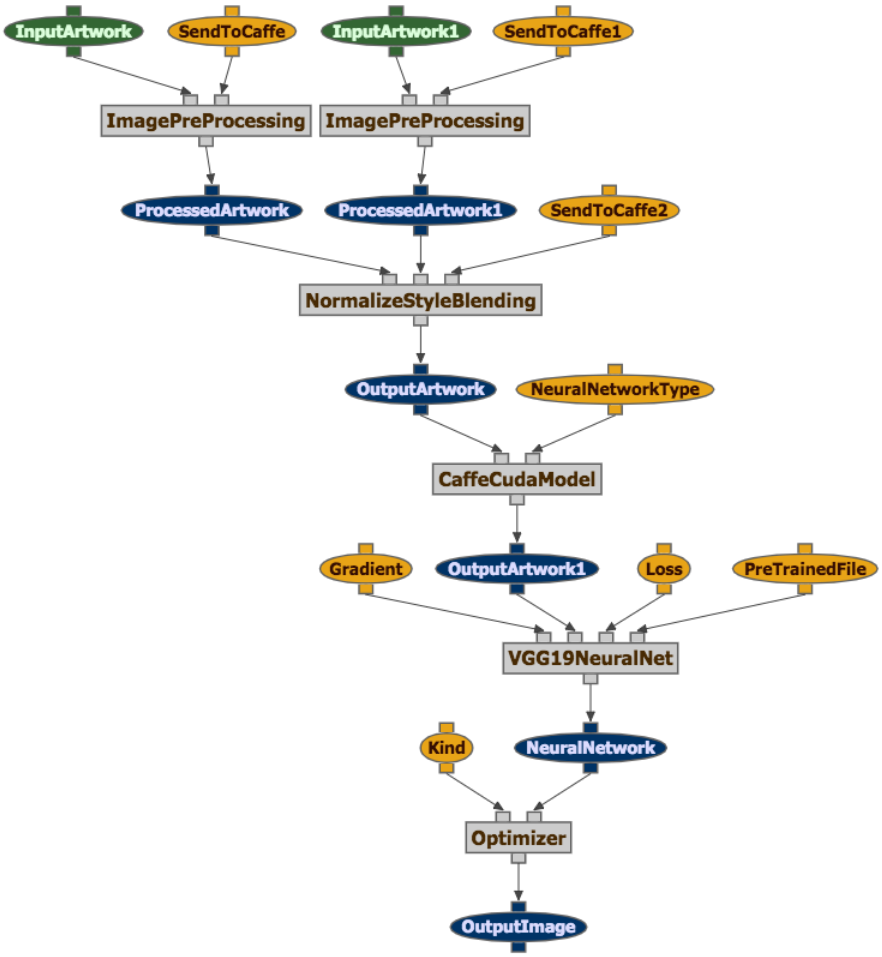


Figure 15: Workflow for the Neural Algorithm of Artistic Style using Lua and Torch.

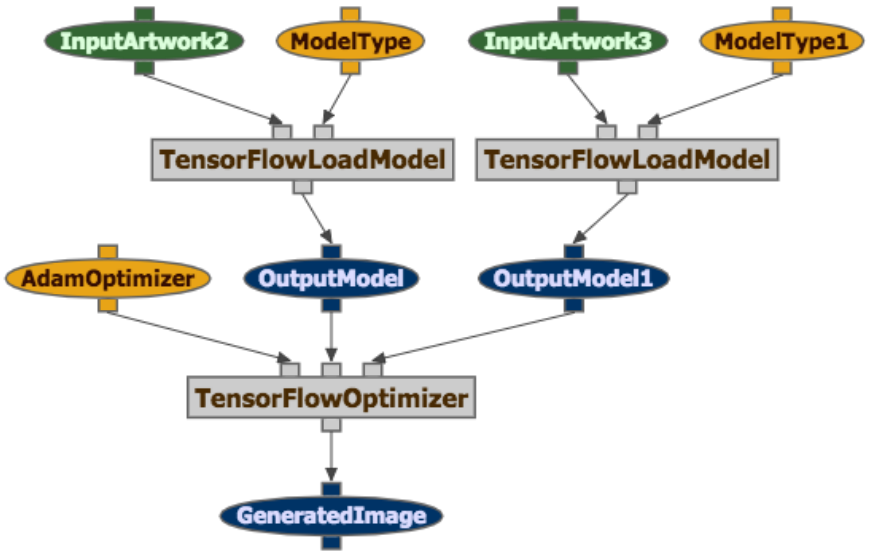


Figure 16: Workflow for the Neural Algorithm of Artistic Style using Python and Google's TensorFlow.

Figure 17: Reproducing the results from [47] using the workflows in Figure 15.

ments to Albert Bierstadt's The Morteratsch Glacier, Upper Engadine Valley, Pontresina 1895 (The Brooklyn Museum, Wikiart[1]), as shown in Figure 18.

CNNs pass filters of shared weights across overlapping image patches to learn convolutions, also called feature maps. Following the example of the paper, we utilize the publicly available and trained VGG network. The method for determining the style is based on texture extractions using correlations between feature-maps within a layer. The final reconstructed image is created by first randomly generating a white-noise image and then using gradient descent for the optimization. In the workflows in Figures 15 and 16, we use Adam Optimization, an algorithm for first-order gradient-based optimization. Finally, the loss function in the article is a weighted sum of the style and content mean squared error loss functions and is used to generate the final new image. We can re-use the Adam Optimization module in the workflow in Figure 16 if we

[1] http://www.wikiart.org/en/albert-bierstadt/the-morteratsch-glacier-upper-engadine-valley-pontresi

Figure 18: Applying the Neural Algorithm for Artistic Style from [47] to the Bierstadt painting using the workflows in Figure 15. The first row combines the Bierstadt painting with the styles of Van Gogh, Munch, and Escher. The second row combines the Bierstadt painting with the styles of Frida, Matisse, and Picasso.

did not want to use the optimization provided in TensorFlow.

9. Analysis of Time/Work Savings

In the implementation of the multimedia analysis task in Section 5, we incorporated the original crawler and extracter into WINGS and then added on various text analysis and image analysis workflow fragments, including fusing their results and adding components to help visualize the results. This involved writing simple component wrapper scripts for both of the existing python scripts and setting up the mySQL database interface. The original development of the python version of the crawler/extracter had taken several months; this was quite involved as appropriate algorithms had to be researched, in addition to developing the code. The original crawler and extracter were componentized into WINGS components, as shown in Figure 4.

This process took roughly two days as the original programs had to be made independent of the original development environment, account for supporting libraries, and had to interface with the external Database system that was distributed on the Web. Once this was done, the extension of the other components via workflow fragments for Image Analysis, Text Analysis, and their Fusion and visualization, took approximately one day, saving effort estimated to be on the order of 300 person-hours of work. This was estimated by the original developers using one postdoc and one graduate student working at a similar pace as in the development of the original prototype as they worked to identify appropriate algorithms for image and text analysis, implement them and incorporate them into their nascent crawler/extracter prototype, and then investigated a fusion methodology as well as the tools to visualize and analyze the results.

For the Text Analysis task in Section 6, We recruited three high school students with limited programming background to use our system over a period of a week. The students had taken two semesters of introduction to programming in the eight and ninth grades, and were entering tenth grade in the coming year. After a short tutorial, they were then were asked to formulate useful tasks for themselves that would require running workflows or extending them by adding new components. During the five days, the students did the following tasks:

- Became familiar with workflows as a software paradigm
- Learned to use the system and run simple workflows to analyze data (e.g., compare sets of html files to see how they would be classified)
- Learned to use pre-existing workflows for advanced text analytics (e.g., run workflows for document clustering and topic detection and compare their performance for different threshold parameters)
- Extended existing workflows with new workflow components that they developed

- Analyzed twitter data to detect topic trends by applying pre-existing advanced text analytic workflows

Reusing the workflow fragments also allowed time savings and collaboration between geographically dispersed colleagues, postdocs, graduate students, undergraduate students, and even high school students. For example, the high school students in Section 6 were able to collaborate with researchers locally (in Los Angeles, CA) as well as researchers that worked on creating some of the initial workflow fragments and were located in Germany. Similarly, for the Image and Video Analysis tasks, one graduate student developed and marked the workflows and dataset for image analysis (in Los Angeles, CA) while another undergraduate student (in Austin, TX) used the pre-existing fragments and developed more image analysis fragments. Researchers (in Fitchburg, MA) added additional image analysis algorithms, including the neural algorithm for artistic style. These fragments, as well as the previous image analysis components, were then utilized by humanities students and scholars (in Boston, MA) to conduct analyses of artworks, inlcuding the Bierstadt painting above. Creation of the artistic style fragments required approximately 25 person-hours and considerable expertise but using them only required nominal training in the use of the WINGS framework, which on average was completed in a two-hour session.

For the Activity Recognition tasks in Section 7, we were able to re-use workflows for fundamental image and video analysis to avoid doing separate implementations, resulting in time and personnel savings. The workflows framework also allowed for quick deployment of both the Atomic Pair Actions and Atomic Group Actions datasets, including the comparison of various approaches and algorithms. Finally, in the Neural Algorithm Analysis of Artistic Style task in Section 8, we were able to use some of the workflows developed earlier for video activity recognition, as well.

10. Conclusions

This kind of re-use and re-purposing of workflow fragments across different data domains can be generalized to other scientific fields and allows scientists to link across different disciplines [2, 5]. For example, in geosciences, researchers observe the surface of the Earth at criticial points and examine moisture levels from above and below. This, in turn, depends closely on weather models, models of soil, rain, etc. Thus, they also need to use approaches from many different disciplines to analyze data from multiple domains. Examples such as this also abound in Biology [5], particularly in Proteomics and Genomics.

In this work, we only illustrate re-use of our workflow fragments by ourselves and our collaborators, not by third parties [6]. However, this paper does illustrate the potential for reuse of workflow fragments and, if they are shared with other researchers, more scientists can use such workflow fragments in their own applications instead of having to re-implement them or, even worse,

forego such an analysis. One of the issues involved with sharing workflow fragments is the open question of how to describe them so they are re-usable by others. We intend to examine this in future work where one promising approach is to use a Component Ontology by function as an aspect [48]: i.e., being able to find workflow fragments according to a user query to search for a specific kind of component that is retrieved for the user; we can also find workflow fragments by typing about them in English [49].

11. Acknowledgments

This research was supported in part by the US National Science Foundation (NSF) with grant numbers ICER-1440323, IIS-1344272, and ACI-1355475, in part under grant #1019343 to the Computing Research Association for the CIFellows Project, and in part under the National Endowment for the Humanities (NEH) Grant under Award HD-248360-16. We would like to thank Taylor Alarcon, Catherine A. Buell, Alyssa Deng, Matheus Hauder, Hyunjoon Jo, Yan Liu, Andrew Philpot, and William P. Seeley for their discussions and their assistance in developing some of the workflow fragments.

[1] Y. Gil, V. Ratnakar, J. Kim, P. A. Gonzalez-Calero, P. Groth, J. Moody, E. Deelman, Wings: Intelligent workflow-based design of computational experiments, IEEE Intelligent Systems 26 (1).
URL papers/gil-etal-ieee-is-11.pdf

[2] Y. Gil, P. Szekely, S. Villamizar, T. Harmon, V. Ratnakar, S. Gupta, M. Muslea, F. Silva, C. Knoblock, Mind your metadata: Exploiting semantics for configuration, adaptation, and provenance in scientific workflows, in: Proceedings of the Tenth International Semantic Web Conference, 2011.

[3] Y. Gil, V. Ratnakar, E. Deelman, G. Mehta, J. Kim, Wings for pegasus: Creating large-scale scientific applications using semantic representations of computational workflows, in: AAAI, 2007, pp. 1767–1774.
URL papers/gil-etal-iaai07.pdf

[4] Y. Gil, P. Groth, V. Ratnakar, C. Fritz, Expressive reusable workflow templates, in: Proceedings of the Fifth IEEE International Conference on e-Science (e-Science), Oxford, UK, 2009.
URL papers/gil-etal-escience09.pdf

[5] T. M. Kurc, S. Hastings, V. S. Kumar, S. Langella, A. Sharma, T. Pan, S. Oster, D. Ervin, J. Permar, S. Narayanan, Y. Gil, E. Deelman, M. W. Hall, J. H. Saltz, High performance computing and grid computing for integrative biomedical research, Journal of High Performance Computing Applications 23 (3) (2009) 252–264.
URL papers/kurc-et-al-hpc09.pdf

[6] A. Goderis, U. Sattler, P. Lord, C. Goble, Seven bottlenecks to workflow reuse and repurposing, in: The Semantic Web, Springer Berlin / Heidelberg, 2005, pp. 323–337.
URL http://dx.doi.org/10.1007/11574620_25

[7] D. D. Roure, C. Goble, S. Aleksejevs, S. Bechhofer, J. Bhagat, D. Cruickshank, D. Michaelides, D. Newman, The myexperiment open repository for scientific workflows, in: Open Repositories 2009, 2009, event Dates: May 2009.
URL http://eprints.soton.ac.uk/267131/

[8] K. Wolstencroft, P. Fisher, D. D. Roure, C. Goble., Research in a Connected World, 2009, Ch. Scientific Workflows.

[9] D. Garijo, O. Corcho, Y. Gil, B. A. Gutman, I. D. Dinov, P. Thompson, A. W. Toga, Fragflow: Automated fragment detection in scientific workflows, in: Proceedings of the IEEE Conference on e-Science, Guarujua, Brazil, 2014.

[10] D. Garijo, O. Corcho, Y. Gil, Detecting common scientific workflow fragments using templates and execution provenance, in: Seventh ACM International Conference on Knowledge Capture (K-CAP), Banff, Canada, 2013.

[11] M. Hauder, Y. Gil, R. Sethi, Y. Liu, H. Jo, Making data analysis expertise broadly accessible through workflows, in: In Proceedings of the Sixth Workshop on Workflows in Support of Large-Scale Science (WORKS), held in conjunction with SC, 2011.
URL papers/hauder-etal-works11.pdf

[12] E. Deelman, G. Singh, M. H. Su, J. Blythe, Y. Gil, C. Kesselman, G. Mehta, K. Vahi, G. B. Berriman, J. Good, A. Laity, J. C. Jacob, D. S. Katz, Pegasus: a framework for mapping complex scientific workflows onto distributed systems, Scientific Programming Journal.

[13] T. Oinn, M. Greenwood, M. Addis, N. Alpdemir, J. Ferris, K. Glover, C. Goble, A. Goderis, D. Hull, D. Marvin, P. Li, P. Lord, M. Pocock, M. Senger, R. Stevens, A. Wipat, C. Wroe, Taverna: lessons in creating a workflow environment for the life sciences, Concurrency and Computation: Practice and Experience.

[14] S. P. Callahan, J. Freire, E. Santos, C. E. Scheidegger, C. T. Silva, H. T. Vo, Vistrails: Visualization meets data management, ACM SIGMOD.

[15] B. Ludaescher, I. Altintas, C. Berkley, D. Higgins, E. Jaeger, M. Jones, E. A. Lee, J. Tao, Y. Zhao, Scientific workflow management and the kepler system, Concurrency and Computation: Practice and Experience.

[16] M. Reich, T. Liefeld, J. Gould, J. Lerner, P. Tamayo, J. Mesirov, Genepattern 2.0, Nature Genetics.

[17] I. Taylor, E. Deelman, D. Gannon, M. Shields, Workflows for e-science, Springer Verlag.

[18] J. Kim, Y. Gil, M. Spraragen, Principles for interactive acquisition and validation of workflows, Journal of Experimental and Theoretical Artificial Intelligence.

[19] Y. Gil, V. Ratnakar, C. Fritz, Assisting scientists with complex data analysis tasks through semantic workflows, in: Proceedings of the AAAI Fall Symposium on Proactive Assistant Agents, Arlington, VA, 2010.

[20] D. Garijo, Y. Gil, O. Corcho, Abstract, link, publish, exploit: An end-to-end framework for workflow sharing, Future Generation Computing Systems.

[21] I. J. Taylor, E. Deelman, D. B. Gannon, M. Shields, Workflows for e-Science: Scientific Workflows for Grids, Springer-Verlag, 2006.

[22] M. Hauder, Y. Gil, Y. Liu, A framework for efficient text analytics through automatic configuration and customization of scientific workflows, in: In Proceedings of the Seventh IEEE International Conference on e-Science, Stockholm, Sweden, 2011.
URL papers/hauder-etal-eScience2011.pdf

[23] Y. Gil, V. Ratnakar, R. Verma, A. Hart, P. Ramirez, C. Mattmann, A. Sumarlidason, S. L. Park, Time-bound analytic tasks on large datasets through dynamic configuration of workflows, in: Proceedings of the Eighth Workshop on Workflows in Support of Large-Scale Science (WORKS), held in conjunction with SC 2013, Denver, CO, 2013.

[24] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, S. J. Cunningham, Weka: Practical machine learning tools and techniques with java implementations (1999).

[25] G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools.

[26] A. K. McCallum, Mallet: A machine learning for language toolkit (2002).
URL http://mallet.cs.umass.edu

[27] L. Moreau, B. Clifford, J. Freire, J. Futrelle, Y. Gil, P. Groth, N. Kwasnikowska, S. Miles, P. Missier, J. Myers, B. Plale, Y. Simmhan, E. Stephan, J. V. denBussche, The open provenance model core specification (v1.1), Future Generation Computer Systems 27 (6).
URL papers/moreau-etal-fgcs11.pdf

[28] L. Moreau, P. Missier, K. Belhajjame, R. B'Far, J. Cheney, S. Coppens, S. Cresswell, Y. Gil, P. Groth, G. Klyne, T. Lebo, J. McCusker, S. Miles, J. Myers, S. Sahoo, C. Tilmes, Prov-dm: The prov data model, world Wide Web Consortium (W3C) (2013).
URL http://www.w3.org/TR/prov-dm/

[29] Y. Yang, J. O. Pedersen, A comparative study on feature selection in text categorization, Morgan Kaufmann Publishers, 1997, pp. 412–420.

[30] P. Turaga, R. Chellappa, V. S. Subrahmanian, O. Udrea, Machine recognition of human activities: A survey, CSVT.

[31] J. Aggarwal, M. Ryoo, Human activity analysis: A review, ACM Computing Surveys.

[32] B. Song, R. J. Sethi, A. K. Roy-Chowdhury, Robust Wide Area Tracking in Single and Multiple Views, in: T. B. Moeslund, L. Sigal, V. Krüger, A. Hilton (Eds.), Visual Analysis of Humans, Springer-Verlag, 2011, pp. 1–18. doi:10.1007/978-0-85729-997-0.
URL http://www.ee.ucr.edu/~amitrc/LAPbook_tracking.pdf

[33] N. M. Nayak, R. J. Sethi, B. Song, A. K. Roy-Chowdhury, Motion Pattern Analysis for Event and Behavior Recognition, in: T. B. Moeslund, L. Sigal, V. Krüger, A. Hilton (Eds.), Visual Analysis of Humans, Springer-Verlag, 2011, pp. 289–309. doi:10.1007/978-0-85729-997-0.

[34] J. Aggarwal, Q. Cai, Human motion analysis: A review, CVIU.

[35] R. J. Sethi, H. Jo, Y. Gil, Structured analysis of the isi atomic pair actions dataset using workflows, Pattern Recognition Letters SI:SAHAR.

[36] R. J. Sethi, Towards defining groups and crowds in video using the atomic group actions dataset, in: IEEE International Conference on Image Processing (ICIP), 2015.

[37] Y. Wang, Y. Tian, L. Duan, Z. Hu, G. Jia, ESUR: A system for Events detection in SURveillance video, in: ICIP, no. 60973055, IEEE, 2010, pp. 2317–2320.
URL http://www.jdl.ac.cn/doc/2010/ICIP2010_ESURASYSTEMFOREVENTSDETECTIONINSURVEILLANCEVIDEO.pdf

[38] D. Zhang, D. Gatica-Perez, S. Bengio, Modeling individual and group actions in meetings with layered HMMs, IEEE Transactions on Multimedia 8 (3) (2006) 509–520.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1632036

[39] R. J. Sethi, A. K. Roy-Chowdhury, Physics-based Activity Modelling in Phase Space, in: ICVGIP, 2010.

[40] R. J. Sethi, A. K. Roy-Chowdhury, Modeling and Recognition of Complex Multi-Person Interactions in Video, in: ACM MM MPVA, 2010, pp. 0–3.

[41] N. M. Oliver, B. Rosario, A. P. Pentland, A Bayesian computer vision system for modeling human interactions, PAMI 22 (8) (2000) 831–843.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=868684

[42] N. Oliver, B. Rosario, A. P. Pentland, A Bayesian computer vision system for modeling human interactions, in: ICVS, Springer-Verlag, 1999.
URL http://www.springerlink.com/content/cc88qcwxx9u3v5ka/?MUD=MP

[43] B. Ni, S. Yan, A. Kassim, Recognizing Human Group Activities with Localized Causalities, in: CVPR, 2009, pp. 1470–1477.
URL http://www.lv-nus.org/publics.html

[44] Y. Zhou, B. Ni, S. Yan, T. S. Huang, Recognizing pair-activities by causality analysis, ACM Transactions on Intelligent Systems and Technology 2 (1) (2011) 1–20. doi:10.1145/1889681.1889686.
URL http://dl.acm.org/citation.cfm?id=1889686

[45] R. J. Sethi, L. Bry, The madsci network: Direct communication of science from scientist to layperson, in: 21st International Conference on Computers in Education (ICCE), 2013.

[46] H. Jo, K. Chug, R. J. Sethi, A Review of Physics-based Methods for Group and Crowd Analysis in Computer Vision, Journal of Postdoctoral Research 1 (1) (2013) 4–7.

[47] L. A. Gatys, A. S. Ecker, M. Bethge, A neural algorithm of artistic style (Aug 2015).
URL http://arxiv.org/abs/1508.06576

[48] R. Bergmann, Y. Gil, Retrieval of semantic workflows with knowledge intensive similarity metrics, in: Proceedings of the Nineteenth International Conference on Case Based Reasoning (ICCBR), Greenwich, London, 2011.
URL papers/bergmann-gil-iccbr11.pdf

[49] Y. Gil, V. Ratnakar, C. Fritz, Tellme: Learning procedures from tutorial instruction, in: Proceedings of the ACM International Conference on Intelligent User Interfaces, Palo Alto, CA, 2011.
URL papers/gil-etal-iui11.pdf