Classification of Game Events in Ice Hockey Game Film

Phil Cohn

March 15, 2013

Abstract

We've designed an approach to use player attributes derived from ice hockey game film with simple classification algorithms to determine what game events are depicted in given frames of film. By surveying existing work in the field of image processing we identify a series of extractable location and orientation attributes that describe each visible player in a given frame. To evaluate this approach we use machine learning techniques to try to identify predetermined game events that vary in terms of location dependence and player alignment. Manipulating attributes and introducing noise to the dataset simulates an approach that gathers player attribute information automatically using image processing techniques. We hope to expand our approach to play a primary role in a comprehensive video analysis system capable of recognizing and classifying ice hockey game events automatically for coaching purposes.

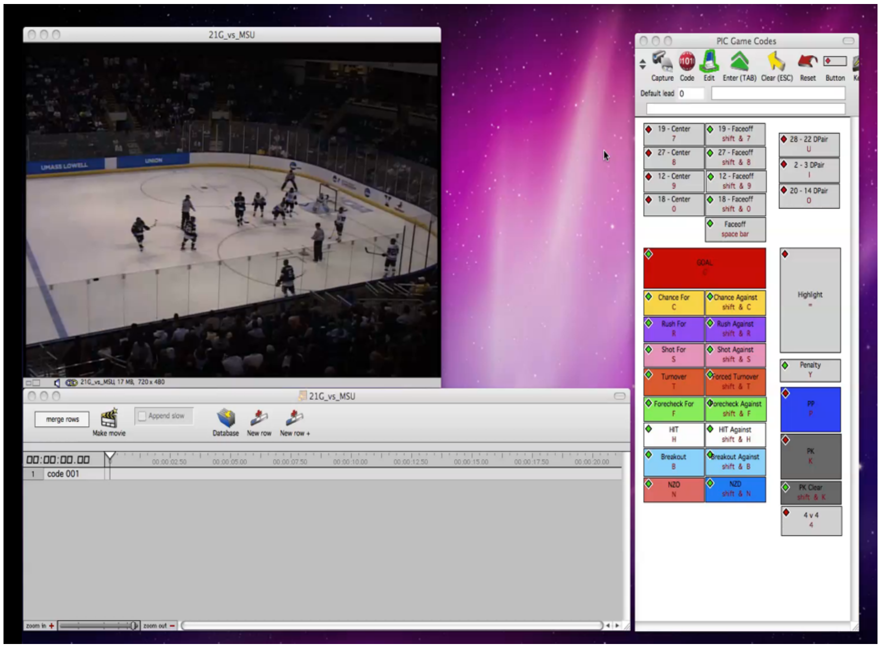


Figure 1: Sportscode Gamebreaker [11] is an example of existing software used for comprehensive video analysis in sports. It employs a window to show the given video feed, a window of customizable code buttons, and a timeline window to show the highlighted segments where they appear in the video file as a whole.

1 Introduction

Video analysis is a central tool for men's ice hockey coaching at amateur and professional levels alike. Game and practice film allows coaches to document the progress of their athletes over time, and in-game analysis enhances a coach's ability to make strategic and personnel adjustments during the course of a game. The objective of in-game analysis is to provide coaches with important data from a given segment of game-play, and to present that data in a way that it can be easily and quickly navigated. Coaches and video analysts use software to 'breakdown' or 'code' game film in live time by tagging segments of game film that are important for coaching purposes. Video analysis systems, such as the example shown in Figure 1, capture video provided by an external camera typically positioned parallel to the center-ice red line at a high vantage point as seen in Figure 6. The video feed shows approximately one zone of the ice at a time. The three zones on the ice are diagrammed in Figure 2, and include the defensive, neutral, and offensive zones based on

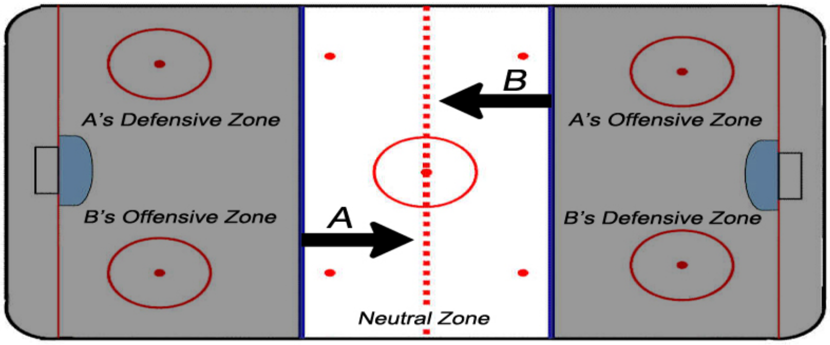


Figure 2: The ice surface is divided into three zones by the two blue lines. The zone in the middle is the neutral zone, and the zones on each side are the offensive/defensive zones depending on team orientation. For example, if Team A is defending the goal on the left side of the diagram, the left-most zone is the defensive zone and the right-most zone is the offensive zone.

team orientation. This view is large enough for strategic analysis and condensed enough to identify player numbers for individual player analysis.

When both teams are at full-strength during a game, there are twelve players on the ice, each with a specific role for his team. There are three forwards on the ice at a given time for each team that skate predominantly forward and play very aggressively in the offensive zone in an effort to score goals. There are two defensemen on the ice for each team that generally play a conservative style. Defensemen often skate backwards as they retreat to the defensive zone to defend their net, and they rarely skate far beyond the offensive blue line. Goaltenders have the most limited range of motion of all of the individuals on the ice. One goaltender for each team inhabits the goal in his team's defensive zone, and moves predominantly side to side within the blue crease to afford the opposing shooter the worst possible shooting angle.

Strategic norms and standard characteristics of player movement for different roles make the general location of each of the players on the ice fairly predictable during certain game-play situations, however it is common for players to be drawn away from their primary territory on the ice. While existing systems are capable of identifying player characteristics within a given video sequence, identifying game events is still a difficult task. Game events can vary in location dependence, the number of players involved, and the alignment of those players on the ice. Variations in each of these factors effect the difficulty inherent in identifying a given event. We propose to use classification techniques to manipulate player location and orientation data to classify ice hockey video frames to represent one of several predetermined game events.

2 Approach Overview

To classify a given video frame as an example of a specific game event we need to identify features about the visible players in the video feed that will tell us something about what is happening in each frame. Ideally we would use the role of each player as a feature for classification, however player role identification is a difficult task. It is common for players to be drawn away from their primary role responsibilities on the ice. For example, in situations when a defenseman joins the play offensively, a forward will commonly shift into the voided defensive position until normal positioning can be reestablished. Existing image processing techniques are capable of extracting useful information from a video feed in live time. With these techniques in mind we can recognize visible features that we can use to describe each player in a given frame of game film. We consolidate these player attributes to describe a single frame instance that we manually label with the game event represented in that frame. Using simple classifiers, we can look at each frame instance and begin to understand what assortment of player attributes best describes each game event.

3 Related Work

Research in the field of image processing provides the foundation for our classification task. We can identify useful information that we can extract from game film for classification by looking at existing work in image processing. First we need a way of identifying players in each frame of game film. Yilmaz et al [12] surveys the field of object tracking, and defines the task to be the problem of estimating the trajectory of an object in the image plane as it moves around a scene. A tracker assigns consistent labels to tracked objects in different frames of video [12]. Choi and Savarese [1] propose an object tracker that is capable of working under very challenging conditions. Their system relies on a framework that simultaneously estimates camera parameters and tracks objects. This allows the system to track multiple objects within a single video sequence and account for unexpected camera motion, which is important to note for our purposes. The given video feed must follow the action of the game which is free-flowing and spontaneous, so we need a tracker that

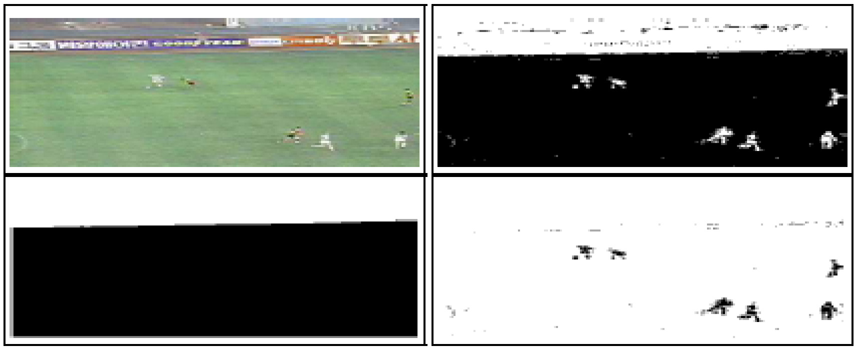


Figure 3: Field extraction is a technique for background estimation that separates moving objects in a sequence from the static background. Yongduek et al [13] present an approach to field extraction and use this technique for tracking players in soccer game film.

can not only keep track of the location of each moving player, but also account the movement of the camera providing the video feed. The given video feed can assumes much less challenging conditions than the video sequences used by Choi and Savarese because they are provided by a stationary camera that simply pans back and forth and does not zoom substantially. Because of this, we are confident that tracking software exists that will be compatible for our task.

Background estimation is an important part of object tracking. Ridder et al [7] introduces a technique for background estimation or field extraction that uses Kalman-Filtering. In most systems the first step in tracking objects is to separate the foreground from the background to help detect motion. Ridder et al's approach assumes that objects in the foreground are moving continuously because the foreground cannot be detected if there is no motion between frames [7]. Motion is detected by evaluating changes in pixel values in subsequent frames. If objects in the foreground are not in continuous motion, there will be no changes in pixel value between frames and no motion will be detected. Figure 3 shows an example of how Yongduek et al use field extraction for analysis of soccer game film. Background estimation is not a very difficult task for ice hockey game film because the ice surface is consistently white, and because it is very rare that players on the ice will stay stop moving completely.

While object tracking shows the location of each moving player within the frame of the video feed, we want to know the location of each moving player on the ice. To do this we can use landmark detection, which is discussed by Wang and Parameswaran [10]. Landmark detection uses static lines or objects on the ground surface to place moving objects as the camera moves, changing the perspective of the scene. Wang and Parameswaran do landmark detection using soccer game film, and they use the goal mouth, the corners of the field, the field lines, and the center-field circle as useful landmarks. Useful landmarks in ice hockey game film include the face-off circles, the blue and red lines, the two nets, and the blue crease in front of each net. Detecting landmarks usually starts with the most significant features [10]. Landmark detection allows us to draw conclusions about where players are located on the ice with respect to selected landmarks on the ice that do not move. We can use these landmarks as reference as the moving camera changes the perspective of the video feed within which the players visibly move around the ice. To achieve high granularity for player location this tool is very useful.

Once we know where each visible player is located on the ice surface, we want to know what team he is on and further what we can conclude about his current action. Yongduek et al [13] uses histogram distribution to identify similar uniform colors among moving players in soccer game film. A similarity measure is computed using pixel information of all moving players and the two teams are differentiated from one another. Lu and Little [5] propose to use grids of Histograms of Oriented Gradient (HOG) to recognize actions of moving hockey players (Figure 4). The HOG was originally designed for human detection and uses a rectangular tracking region that contains the entire body of a single player. This is important for our purposes because this rectangular region does not sacrifice the discrimination between the foreground and background, which is homogenous in our case. HOG's are not sensitive to changes in uniform color so they can focus solely on shape and orientation of the player rather than the color or texture of his uniform [4]. By precisely isolating the player from the background using an HOG, the player's pose can be analyzed and an action can be assigned.

4 Data

Our dataset is comprised of player instances annotated manually by looking at ice hockey game one frame at a time. The data contains data from two periods of Division I men's hockey game film played at Messa Rink in Schenectady, NY. We use the first period from a game between Union College and Merrimack College,

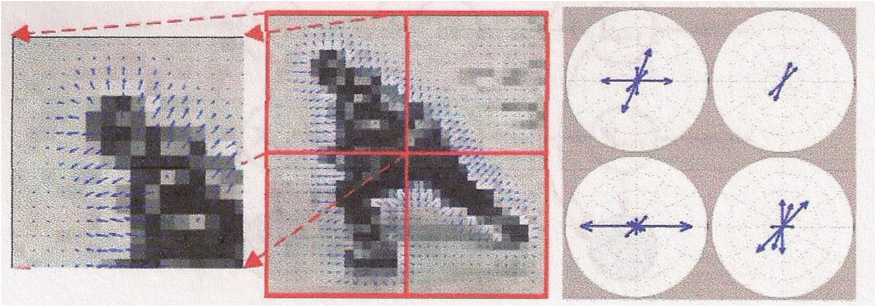


Figure 4: HOG's, introduced by Lu and Little [5] isolate a rectangular region around a moving player and use shape and orientation characteristics to draw conclusions about the current action of the player.

played on October 6, 2012, and the first period from a game played between Union College and Brown University on February 16, 2013. We use image processing techniques to extract and include one frame from every five seconds of gameplay in our dataset, which includes 480 total frames. The periods and games used were selected arbitrarily and were not looked at beforehand.

4.1 Game Events

We will attempt to classify three events common to men's ice hockey. The game events that we choose reflect our objective to provide the classifier with game events that vary in several ways. We choose events that vary in terms of location on the ice, player orientation, the number of players involved in a given event, and the alignment of these players on the ice. Each event presents a different however interesting challenge for classification based on these factors.

4.1.1 Face-offs

The first game event that we will attempt to classify is the face-off. Face-offs mark the start of every game sequence in a hockey game and occur at one of nine face-off dots on the ice. Five of the nine face-off dots are located within a face-off circle, including the four in each of the offensive and defensive zones, in addition to the one at center-ice. The face-off dots are shown in Figure 5. Before a face-off players from each team lineup at the designated face-off circle, the center forward for each team positioned opposite one another in front of the face-off dot. When the official drops the puck directly on the dot, each center attempts to

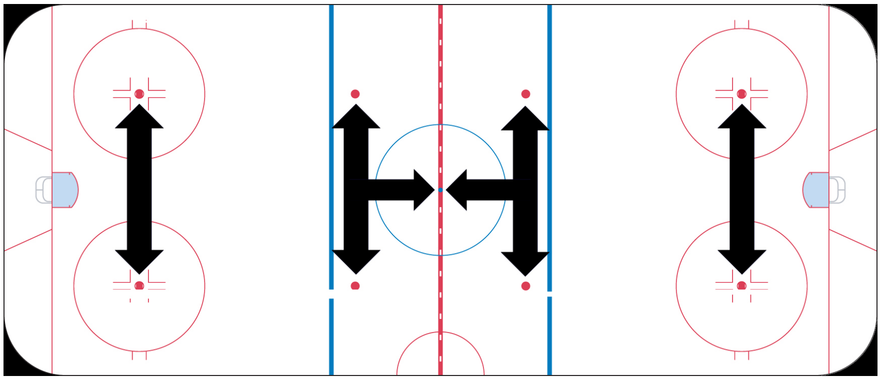


Figure 5: There are nine face-off dots on the ice surface, five of which are located within a face-off circle. The four face-off dots located in the offensive/defensive zones and the face-off dot at center-ice are located inside face-off circles while the four other neutral zone dots are not.

push the puck to one of his teammates. The most common face-off strategy sees the center attempt to pull the puck backwards towards one of his defensemen, assess the current situation and ultimately distribute the puck to a teammate to start an offensive progression. The players must adhere to specific positioning rules before the official drops the puck. Figure 6 shows a face-off at the center-ice face-off dot and Figure 7 show's the same face-off with player roles labeled. The home team players are labeled with red labels, the opposing team with blue labels, and the officials with yellow labels. The players labeled F are forwards, the players labeled D are defensemen, and the individuals labeled O are officials. We can assume that the goalie for each team is positioned in his standard location in the net, therefore each cannot be seen in the given frame.

Face-offs present an interesting challenge as a game event for classification. We can predict that face-offs will not be particularly difficult to classify because the positioning regulations provide for a standard arrangement of players in each frame showing a face-off. However, we must account for how the learner will be able to account for classifying face-offs taken from face-off dots in different locations of the ice. While players may line up for a face-off in a similar formation in both the neutral zone and the offensive zone, the difference in location could certainly cause the classifier to struggle to categorize the events together. Player orientation will likely be useful for classification of face-offs because face-off alignment regulations force all of the players to face their offensive zone. The results for classifying frames as face-offs will tell us a lot about

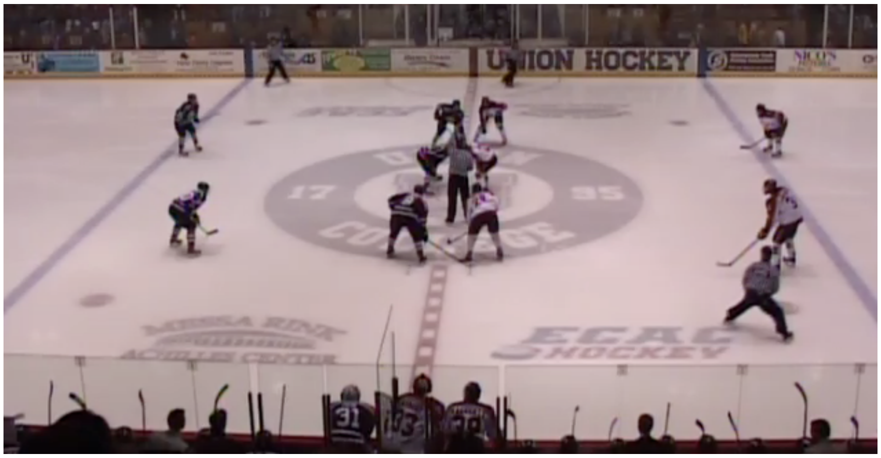


Figure 6: A face-off starts play at the beginning of a period and resumes play after any stoppage. Face-offs are taken on one of nine face-off dots painted on the ice surface, and players are required to adhere to specific alignment regulations before the puck is dropped by the official.

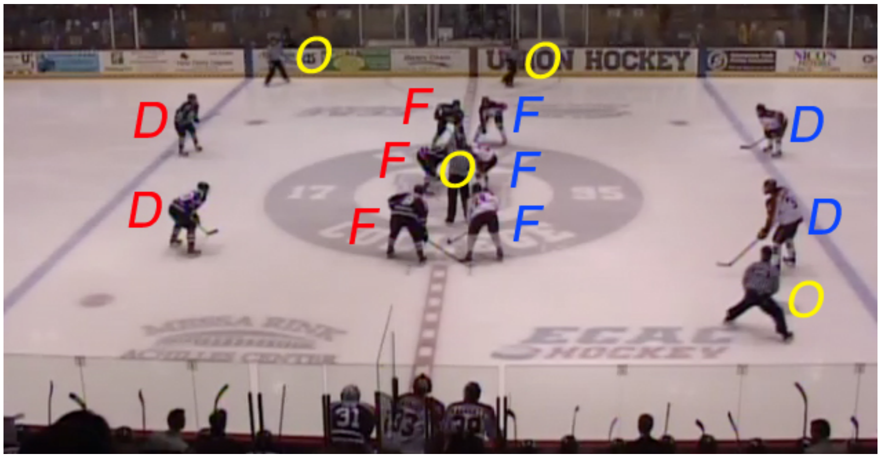


Figure 7: Similar to Figure 6 but in this figure the role of each player and official is marked. The forwards line up facing one another along the center red line, while the defensemen are positioned defensively in the back of the neutral zone. While one official drops the puck the other officials are positioned along the perimeter of the neutral zone.

the importance of player location and the specificity of player location as features for classification purposes, in addition to presenting a comparison of whether orientation or location attributes are more important to accurate classification.

4.1.2 Breakouts

A breakout is a common game event that occurs in one team's defensive zone. The team in their defensive zone possesses the puck and attempts to carry the play into the neutral zone to begin an offensive sequence. Depending on the alignment and pressure applied by the opponent, breakouts can be very strategically structured or very spontaneous. Structured breakouts commonly begin after each team has completed a line-change. A defenseman typically holds the puck behind the net in his defensive zone until each team changes players. Once all of the appropriate players are on the ice, the team that possesses the puck attempts to move the puck into the neutral zone by carrying it or by a sequence of passing. Figure 8 shows an example of a breakout where the defensive team in dark is applying very little pressure to the team carrying the puck in white. The defenseman carrying the puck in the defensive zone has two passing options within the zone in addition to a fair amount of un-defended space to carry the puck out of the zone himself.

Breakouts also present an interesting challenge for classification. We know that a breakout can only occur as a player carries the puck from within his own defensive zone. Because of this, we can be far more dependent on location information features to classify breakouts than we can to classify face-offs. However, while breakouts tend to have a fairly similar structure of players, there is no concrete number of players that are visible on-frame during a breakout. Additionally, we must account for the varying possible alignments of the defensive team during a breakout. While Figure 8 shows an example of a breakout where the defensive team is fairly passive, it is common for opponents to take a much more aggressive approach. Pressuring the team possessing the puck makes it difficult for them to carry the play into the neutral zone.

4.1.3 Scoring Chances

Scoring chances occur when the offensive team plays the puck towards the net in an attempt to score a goal. Most often scoring chances occur when the offensive team records a shot-on-goal (SOG). A shot-on-goal is any shot taken by the offensive team that would go in the net if the opponent goalie was not in the

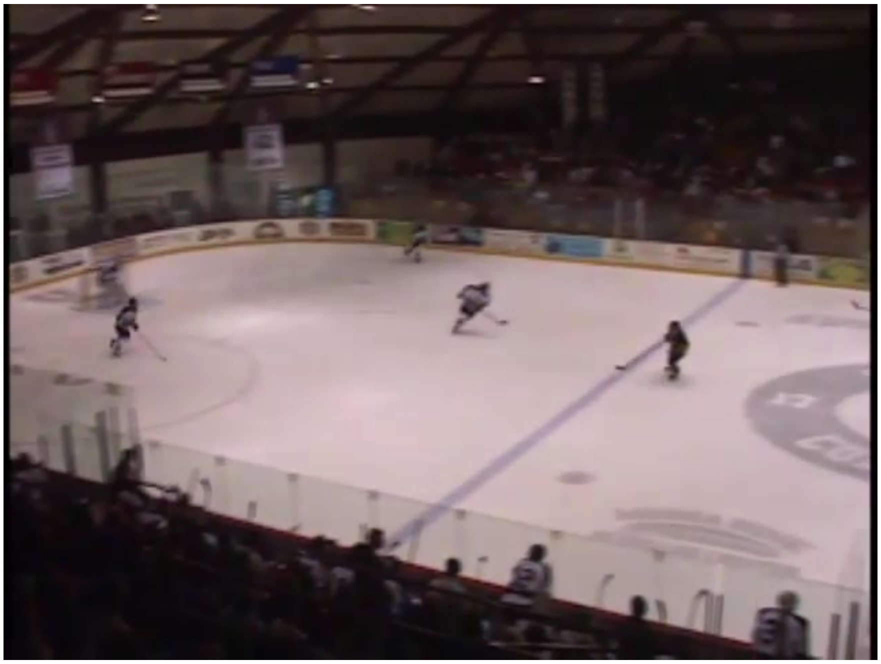


Figure 8: A breakout is the sequence of play during which a team possesses the puck in its defensive zone and attempts to carry the puck into the neutral zone to begin an offensive push. he defenseman carrying the puck in the defensive zone has two passing options within the defensive zone in addition to a fair amount of open space to carry the puck out of the zone himself.

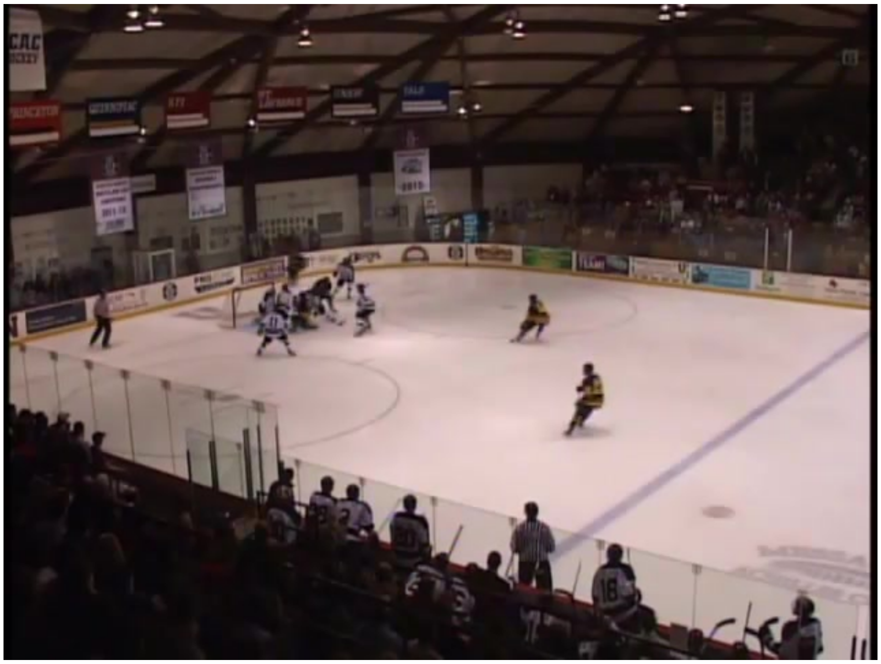


Figure 9: A scoring chance represents a good opportunity for the offensive team to score a goal. Scoring chances always occur near the net, however they are inconsistent in terms of the number of players involved in the play.

way to stop it. Not all SOG's are classified as scoring chances, because they are not all necessarily good opportunities to score. For example, a routine shot from the blue line with little traffic in front of the net is not considered to be a scoring chance, even though it is considered to be a SOG. Conversely, if a player is passed the puck directly in front of the net and fails to direct it on goal, this is a scoring chance even though a SOG is not recorded. Scoring chances can include any number of players on both sides of the puck, and can feature many different alignments of players. Generally, scoring chances have a flexible definition because coaches with different perspectives may disagree on whether a particular game event should be classified as a scoring chance. This discrepancy will make classifying scoring chances a difficult task. An example of a scoring chance can be seen in Frame 9. The dark team is swarming around the white team's net and the white goalie is forced to make a save. There is a lot of activity around the net-front and the defensemen on the dark team can be seen joining the play by moving towards the net from their normal position on the blue line in the offensive zone.

4.2 Features

To classify each frame as a game event we will use a series of player location and orientation attributes. We will look at player location, what direction each player is facing, and whether each player is skating forwards or backwards. While we will extract this information manually from static frames, existing techniques in the field of image processing are capable of extracting this information from a video feed in live time.

4.2.1 Location

We will use two levels of granularity to recognize the location of each player in each frame. We use two levels of granularity because we want to understand the importance of player location specificity for game event classification. The first attribute of location information is simply the zone on the ice in which the player is located. As described earlier and shown in Figure 2, the three zones on the ice include the defensive, neutral, and offensive zones respectively. The zone location for each player is recorded based on that player's team orientation. The video feed shows approximately one zone at any given time. In most situations all of the players visible in the given frame will be in the same zone. While this data will tell us what zone each player is in, we will not have a good idea of the arrangement of players within the zone. In spite of the vague nature of this attribute, we predict that zone information will play an important role in classifying breakouts and scoring chances. Both of these game events occur only in the end-zones of the ice, and the zone attribute will tell us when a given frame shows events in these locations.

In addition to zone information, we will achieve higher granularity by using a gridded representation of the ice surface. The grid is divided into twenty-four rectangular cells on the ice (See Figure 10). The grid consists of eight columns and three rows of cells. The columns are divided by the painted lines on the ice, in addition to vertical lines drawn tangent to the face-off circles in the offensive/defensive zones. Because the grid cells do not intersect zones, zone information is therefore inherent in the grid-cell attribute annotated for each player. The rows of the grid are divided by two parallel horizontal lines that intersect the face-off dots on each side of the ice surface. With this system we are capable of identifying not only what zone each player is in, but also what part of the zone the player is in. We will have a much better idea of how the players on each team are arranged within each frame of game film. With two degrees of location granularity

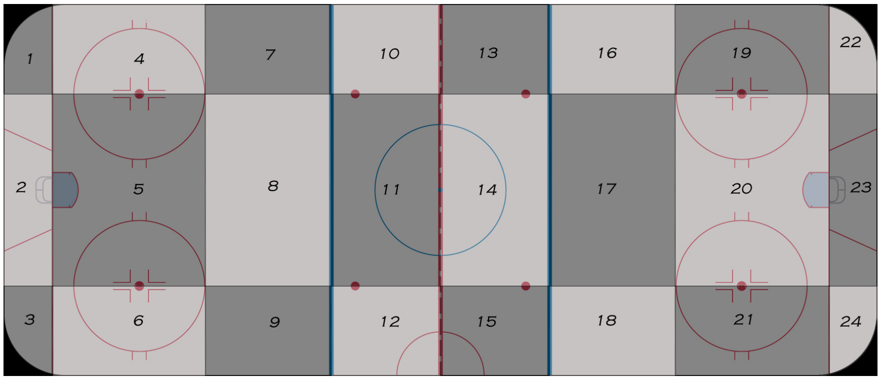


Figure 10: The gridded rink diagram allows us to classify player location with higher granularity than simply using zone information. The numbered cells are not orientation dependent, so squadrons 1-9 represent the defensive zone, 10-15 represent the neutral zone, and 16-24 represent the offensive zone for all players.

we will be able to assess the importance of player location as an attribute for classification.

4.2.2 Direction

The direction that each player is facing is also an important feature for classification. We specify two options for skating direction, including O and D. Players are annotated to be facing either their offensive zone or defensive zone. Team orientation does not effect this attribute. This feature will be especially important for our purposes because we are using individual frames for classification rather than a sequence of frames. While we cannot deduce the exact trajectory of a moving player using a single frame, we can use the direction that the player is facing with the player's skating style to make a determination about what direction the player is going.

4.2.3 Skating Style

Whether a player is skating forwards or backwards can tell us a lot about the players intentions, and in turn we can draw conclusions about what game event is occurring in a given frame. Players skate predominantly forwards in ice hockey. Defensemen skate backwards far more often than forwards and goaltenders. Players skate backwards primarily to retreat towards their own defensive zone to defend the opponent from making a play towards their net. Skating backwards allows defensemen to angle the opponent away from the middle of

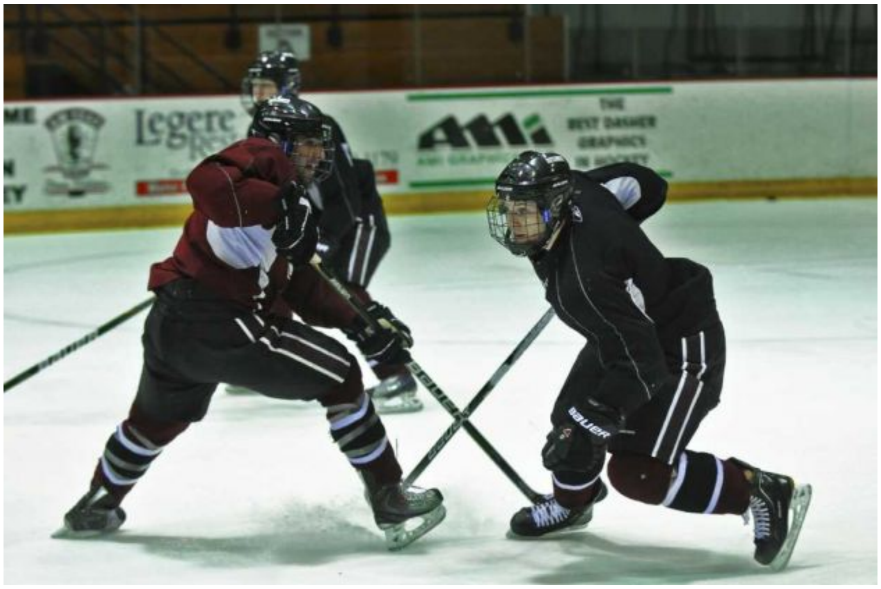


Figure 11: Whether a player is skating forwards is an important feature. Andrew Buote '11 is on the left skating forwards and the Brock Matheson '11 is on the right is skating backwards.

the ice and towards the boards. This affords the defensemen much better defensive positioning than skating forwards, as it allows the defensemen to keep the opponent in front of him at all times and gives him the option to either play the body or play the puck. Figure 11 shows two former Union College players Brock Matheson '11 and Andrew Buote '11. Buote is positioned on the left in the garnet jersey, and is skating forwards preparing to accept a pass. Matheson is positioned on the right in the black jersey, and is skating backwards defending the impending passing lane.

When we are attempting to classify game events we can use skating style information to draw meaningful conclusions about the given frame. It is likely that there will only be one team with players that are skating backwards in a given frame. The players on the offensive team that possesses the puck will skate forwards as they attempt to advance the puck towards the offensive zone and ultimately towards the net to try to create a scoring chance. The defensive team in a given frame will likely have players skating backwards, and this is a good indication that this is the defensive team. We can also use skating style to make assessments about the role of each visible player in the frame. Once we know the role of each player we can more accurately assess each frame as a whole as a possible game event.

4.3 Annotation

To create our dataset we need to consolidate the player attributes that describe each visible player in a given frame into a single frame instance. Each frame instance has forty-eight player attributes that are annotated manually by hand. For each visible player in a given frame, we annotate each of the four attributes that we have introduced, including *Zone*, *Grid-Cell*, *Direction*, and *Skating Style*. The maximum number of visible players in each frame is twelve, including six from each team. We designate twenty-four attributes to represent the six players on the ice for the home team and twenty-four attributes to represent the six players on the ice for the visiting team. We annotate attributes for players on the home team first. Players are added to the dataset in order of their location on the ice. The first player added is the player that is located closest to his defensive end of the ice. The rest of the players are added in order following this rule. Once all of the players on the home team are annotated and added to the frame instance, the players on the visiting team are added in the same manner.

Each player attribute has several options for annotation. For the *Zone* attribute, players located in the defensive zone are annotated with *D*, players in the neutral zone are annotated with *N*, and players in the offensive zone are annotated with *O*. For the *Grid-Cell* attribute, the labeled number of the grid-cell within which each player is located, is recorded for this attribute. For the *Direction* attribute, players that are facing their offensive zone are annotated with *O* and players that are facing their defensive zone are annotated with *D*. For the *Skating Style* attribute, players that are skating forward are annotated with *F* and players skating backwards are annotated with *B*. Each of these attributes are recorded with *NA* if a player is not visible in a given frame. We must use *NA* to represent attributes of players that are off-frame because propositional learners require all instances to have the same number of attributes. We could not use these classification methods if the number of attributes for each instance was dependent on the number of visible players in a given frame.

In addition to the forty-eight player attributes, each frame instance has five class attributes, including a single game event attribute and four boolean attributes. The class attributes are manually recorded during the annotation process so that we can test the performance of different classification algorithms using machine learning techniques for evaluation. The game event attribute simply reflects which of the game events that we have introduced is shown by the given frame. If the frame does not show an example of a face-off, breakout, or scoring chance, then the game event is labeled with the class *UK*, or Unknown. Otherwise, the game event attribute is labeled with *FO* for face-off, *BO* for breakout, or *SC* for scoring chance. The boolean attributes are included so that we can look at each frame and test if it represents a specific game event or not. Each of the boolean attributes has only two possible values, *Y* and *N*. If a frame instance shows an example of a face-off, then the face-off attribute is labeled *Y* and the reset of the boolean attributes are labeled *N*. We include one boolean attribute for each of the game events listed, in addition to one attribute to represent all of the *Unknown* frame instances that do not show an example of any of the three game events that we've described.

5 Classification

For our classification task we use a series of classification algorithms to classify game events using our dataset. We manipulate class and player attributes in order to draw conclusions about our data.

5.1 Algorithms

We use three propositional classification algorithms to manipulate our data. A classifier is a function that maps an unlabeled instance to a label using internal data structures [3]. Propositional learners are standard classifiers that are capable of classifying a set of labeled instances based on a series of given features [6]. The three algorithms that we use are decision tree classification, k-nearest neighbor classification, and Naive Bayes classification. Safavian and Landgrebe [9] introduce decision tree classification (*J48*) as the multistage approach to breaking up a complex decision into the union of several simpler decisions. Each node in a decision tree represents a feature in an instance to be classified, and each branch represents a value that the node can assume [6]. Decision tree classification is attractive because it eliminates unnecessary computation and because global complex decision regions can be approximated by the union of simpler local decision regions at various levels of the tree. However, overlap can cause the number of terminals to be much larger than the total number of classes when the number of classes is large.

The Naive Bayes classifier will also be a useful tool for our classification. The learner greatly simplifies learning by assuming that features are independent of class. Rish [8] uses a Monte Carlo approach to study the classification accuracy for several classes of randomly generated problems. Naive Bayes classification works best in cases when features are completely independent and when features are functionally dependent. For this reason we predict that the Naive Bayes will not classify game events very successfully. While the player attributes that we annotate are not inherently dependent on one another, we use different attributes to represent the same basic feature of player information for each individual player. It is fair for us to assume that the capability for a classifier to recognize this relationship will increase classification accuracy. Cover and Hart [2] present the nearest neighbor classification (*IBk*), in which an unclassified sample point is assigned the value of the nearest of a set of previously classified points. We designate the classifier to use the five nearest points for classification. The learner is based on learning by analogy, and is instance-based or lazy in that it stores all of the training samples and do not build a classifier until a new sample needs to be classified [6]. We can predict that this algorithm will perform well because frames representing the same game event will likely sample to similar points.

5.2 All-Event Testing

For all-event testing we use each of the classification algorithms described above to evaluate our data set. We evaluate the ability for each algorithm to identify a given frame in the data set to be a face-off, breakout, scoring chance, or unknown. We remove the boolean attributes from the data set so that the learners must rely only on player attribute information to classify each frame instance.

5.3 Boolean Testing

For boolean testing we attempt to classify individual game events one at a time. By removing the game event attribute and all but one boolean attribute from the data set, we can evaluate the ability of each learner to identify a single game event. For example, we can remove the boolean attributes that represent breakout, scoring chance, and unknown, in addition to the game event attribute. Once these attributes have been removed, the class label for each instance is simply Y if the frame shows a face-off, and N if it does not. With this technique we can manipulate the class attributes in the data set and evaluate the ability of our classification algorithms to identify individual game events in the dataset.

5.4 Attribute Manipulation

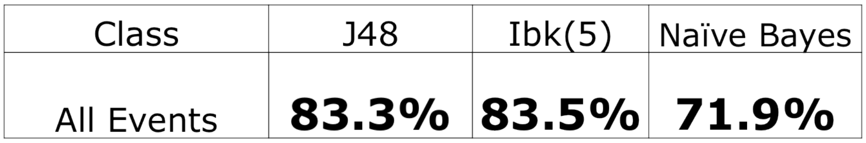
We manipulate the player attributes that each classifier uses for testing to evaluate which attributes contribute most to classification accuracy. For attribute manipulation we use only k-nearest neighbor classification. We do all-event testing using only one of the four player attributes, and further determine the best subset of attributes for all-event and boolean testing using decision tree classification. We can find the best subset of attributes by removing one attribute at a time, and evaluating the classification accuracy without each attribute. We eliminate the attribute that yields the highest accuracy while removed, and then we repeat this process until there remains only one attribute. In addition we add noise to individual attributes to simulate a more realistic confidence in the value of our player attributes.

5.5 Evaluation

To evaluate each of our tests we use a technique called k-fold cross validation. Kohavi [3] claims that to evaluate classification accuracy we want an estimation method with low bias and low variance. We use 10-fold cross validation, which randomly divides the dataset into ten mutually exclusive subsets or folds, of instances. The classifier is trained and tested ten times, each time using a different fold of data for testing, and the remainder of the data for training. The overall classification accuracy of the classifier for the given dataset is the average accuracy calculated using the results from each individual tests. Cross validation lessens the effects of bias because the learner is trained on different data for each test.

6 Results and Discussion

In the following we will present the results of each aspect of our classification experiment. We will discuss possible reasons to explain each result and further draw conclusions from each result.



Class	J48	Ibk(5)	Naïve Bayes
All Events	**83.3%**	**83.5%**	**71.9%**

Figure 12: Decision tree classification (*J48*) and k-nearest neighbor classification are the most accurate learners for classifying each frame as one of our four predetermined game events, however Paired T-Testing shows that the accuracy difference between *J48* and *IBk(5)* is not statistically significant.

6.1 All-Event Testing Results

We are able to classify game events with approximately 83 percent accuracy using decision tree and k-nearest neighbor classification, while Naive Bayes classification yields nearly 72 percent. Results for all-event classification can be seen in Figure 12, and Paired T-Testing shows that the difference in classification accuracy between decision tree classification and k-nearest neighbor classification is not statistically significant. The decision tree that is constructed using *J48* classification to for all-event classification is shown in Figure 13. The root node of the tree is the attribute that represents the grid-cell location of the sixth visible player on the home team. This player is visible in only 55 frame instances in the data set containing 480 frames. For all six players from the same team to be visible we must be looking at a game event occurring in that team's defensive zone. This restriction is true because the goalie cannot leave his defensive zone. The Naive Bayes classifier is not as successful because it assumes that all attributes are independent and unrelated. The difference between the accuracy yielded Naive Bayes classifier and the other two learners is statistically significant.

6.2 Boolean Testing Results

Boolean testing evaluates the capability of each classifier to identify face-offs, breakouts, and scoring chances individually in the data set. Figure 14 shows the results for boolean testing. The highlighted cells represent the highest classification value for each game event evaluated. Naive Bayes performs far better in boolean testing than all-event testing, however it is still less accurate than *J48* and *IBk(5)* for classification of each event. For the boolean classification of breakouts, all three learners achieve a classification accuracy within a five percent range. Decision tree classification reaches the highest accuracy at 90.6 percent. Paired T-Testing

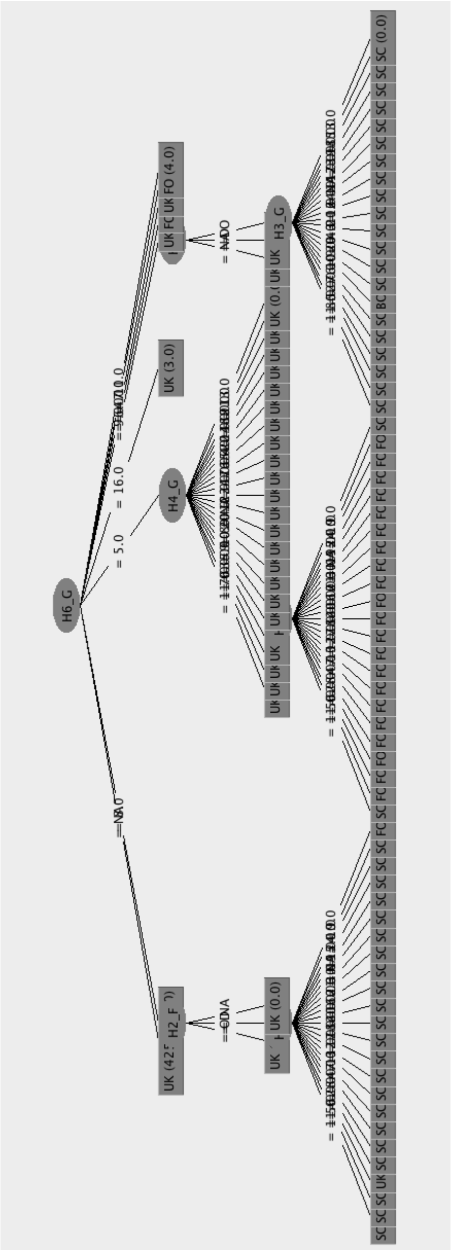
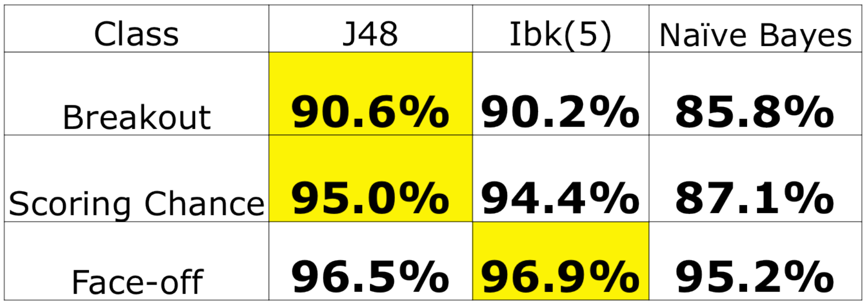


Figure 13: The decision tree constructed to for *J48* all-event classification. The root node is the attribute that grid-cell location of the sixth visible player on the home team, or the player that is located closest to the home team's offensive end of the ice.



Class	J48	Ibk(5)	Naïve Bayes
Breakout	**90.6%**	**90.2%**	**85.8%**
Scoring Chance	**95.0%**	**94.4%**	**87.1%**
Face-off	**96.5%**	**96.9%**	**95.2%**

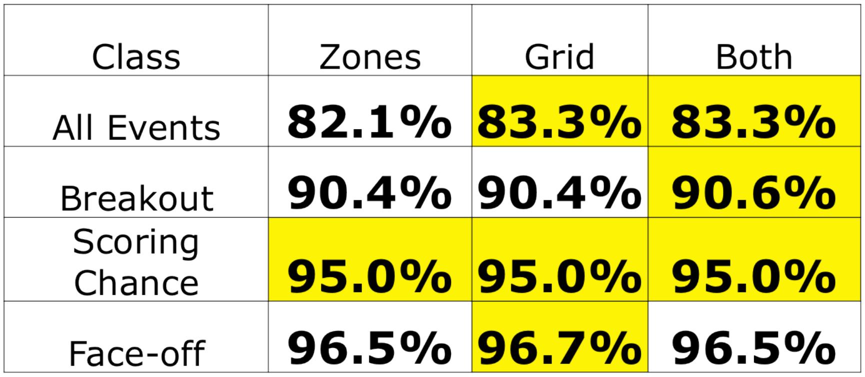
Figure 14: Boolean tests examine the ability for each classifier to identify a single game event in each frame in the dataset. The classification accuracy was generally higher for boolean tests than for all-event tests. The yellow cells in the table represent the highest classification value reached for the game event in that row.

shows that the statistical difference between the results found using each algorithm is not significant. For scoring chances, the 85.8 percent achieved by Naive Bayes proves to be statistically significantly less than the 94 percent accuracy reached using the other two learners. Face-offs are the easiest game event to identify for each classifier, and each classifier reaches at least 95 percent classification accuracy. The statistical difference among the results reached using each algorithm is not significant.

Boolean testing classification results prove to be higher than all-event testing results, showing that identifying a each frame as a single game event is easier than identifying a game event among several possibilities. Face-offs are the easiest game event to identify for each classifier, and we will look at how each attribute contributes to this result in Section 6.4. Breakouts are the most difficult game event to classify, and surprisingly scoring chances do not prove to be a daunting challenge for classification.

6.3 Location Granularity

We evaluate the importance of location granularity by toggling which location attribute we use for classification. The two attributes that we use to represent location are the *Zone* and *Grid-Cell* attributes. Grid-cells do not intersect zone boundaries, so the *Zone* attribute inherently keeps track of zone information. We use decision tree classification for location granularity testing, and we look at classification accuracy yielded using only zone attribute information, only grid-cell attribute information, and using both attributes to describe player location. The results of these tests can be seen in Figure 15. The highlighted cells show the



Class	Zones	Grid	Both
All Events	82.1%	83.3%	83.3%
Breakout	90.4%	90.4%	90.6%
Scoring Chance	95.0%	95.0%	95.0%
Face-off	96.5%	96.7%	96.5%

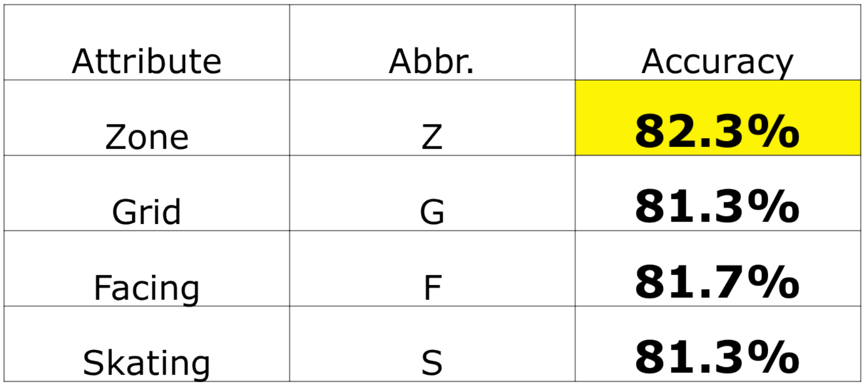
Figure 15: This table shows the difference in accuracy achieved using different degrees of location attribute granularity. The data represented in this table reflects classification completed using decision tree classification. All-event and boolean tests were executed with location attributes toggled.

highest classification accuracy for the all-event test and each of the boolean tests.

Overall, we do not see a statistically significant difference in classification accuracy as we toggle location granularity. For all-event testing, we reach approximately 83 percent using each arrangement of attributes, achieving a slightly higher accuracy using the *Grid-Cell* attribute and both attributes simultaneously. For boolean testing, we reach 90 percent for breakouts, 95 percent for scoring chances, and 96 percent for face-offs. We are surprised to see that classifying game events with only zone location yields approximately the same result as using the grid-cell location. It is possible that our grid-cells do not designate location specifically enough to achieve a better understanding of player location than using only zone information. By increasing the number of grid-cells that cover the playing surface, we can investigate how specific location attribute information must be to contribute to game event classification more effectively.

6.4 Attribute Manipulation

We manipulate player attribute information to draw conclusions about how each attribute that we've chosen contributes to classification accuracy for all-event and boolean testing. We first execute all-event classification with decision tree classification using only one player attribute at a time. We are surprised to find that classifying game events with each attribute individually yields very similar results, approximately 82 percent. The highest classification accuracy is yielded by classifying game events using only the zone location attribute.



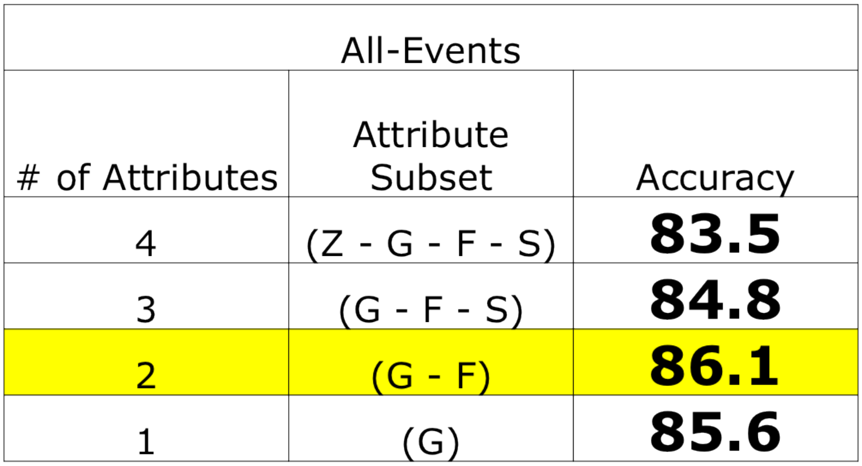
Attribute	Abbr.	Accuracy
Zone	Z	**82.3%**
Grid	G	**81.3%**
Facing	F	**81.7%**
Skating	S	**81.3%**

Figure 16: We run all-event tests using decision tree classification with only one player attribute at a time. Using the zone attribute provided the highest classification accuracy.

The results for this test can be seen in Figure 16. It is interesting to note that with only one attribute we are able to reach a classification accuracy very similar to the accuracy that we reach using all attributes, at approximately 83 percent. This suggests that each attribute has helpful as well as distracting features that contribute to classification accuracy.

After using individual attributes for all-event classification, we further evaluate the effectiveness of each attribute by identifying the best subset of attributes for all-event and boolean classification for each game event. For each test we use k-nearest neighbor classification because it consistently yields high classification accuracy. For all-event classification, the most valuable attribute is the *Grid-Cell* attribute and the best subset of attributes consists of two attributes, including *Grid-Cell* and *Direction*. We achieve 85.6 percent classification accuracy using only the *Grid-Cell* attribute and 86.1 percent with the optimal subset. These results can be seen in Figure 17. The best subset of attributes reaches 86 percent classification accuracy, an improvement of three percent more than is achieved using all four player attributes for classification.

For boolean classification of face-offs, the most valuable attribute is the *Direction* attribute. We classify face-offs successfully at a rate of 97.1 percent using only this attribute. *Direction* is the most important attribute for face-offs because of the alignment regulations that players must adhere to. Players are required to line up for face-offs facing their offensive zone. It would be very unlikely for another scenario during live gameplay to occur that involved every player on the ice facing his offensive zone. For this to occur, every

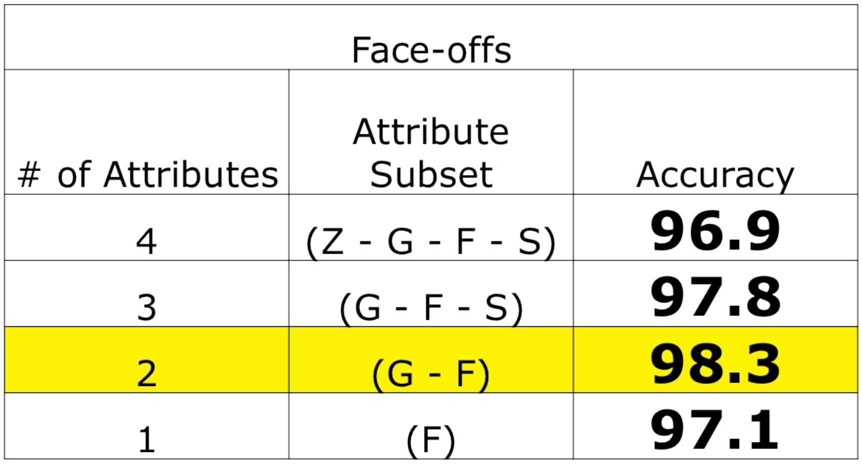


All-Events		
# of Attributes	Attribute Subset	Accuracy
4	(Z - G - F - S)	**83.5**
3	(G - F - S)	**84.8**
2	(G - F)	**86.1**
1	(G)	**85.6**

Figure 17: We evaluate the best subset of attributes for executing all-event testing using k-nearest neighbor classification. The *Grid-Cell* attribute is the most important attribute for classification, and the best subset of attributes includes the *Grid-Cell* attribute and the *Direction* attribute.

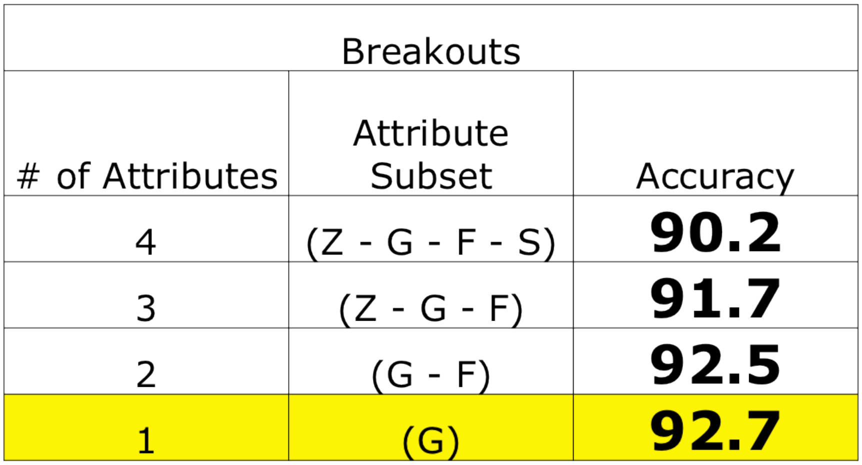
player on both teams would have to be skating forwards towards one another, or an entire team of players would have to be skating backwards. Both these scenarios are highly unlikely to occur, and it is for this reason that *Direction* makes the classification of face-offs very easy for our learners. The best subset of attributes for the classification of face-offs include the *Direction* and *Grid-Cell* attributes, which yield just over 86 percent classification accuracy. The results for this test can be seen in Figure 18.

For boolean classification of breakouts, the most valuable attribute is the *Grid-Cell* attribute. The *Grid-Cell* attribute alone yields the highest classification accuracy of all of the subsets tested at 92.7 percent, however each subset yields an accuracy result of at least 90 percent. The results for his test can be seen in Figure 19. The *Grid-Cell* attribute is the most important attribute for the boolean classification of breakouts because breakouts can only occur in one location on the ice. As described in Section 4.1.2, breakouts must occur in one team's defensive zone as that team attempts to carry the puck up the ice to begin an offensive progression. The alignment of players is fairly standard for a breakout and the *Grid-Cell* attribute is capable of mapping this alignment. The *Grid-Cell* attribute is also the most important attribute for boolean testing of scoring chances. As shown in Figure 20, the best subset of attributes for classifying scoring chances includes the *Grid-Cell* attribute and the *Skating Style* attribute, which yield just over 95 percent accuracy



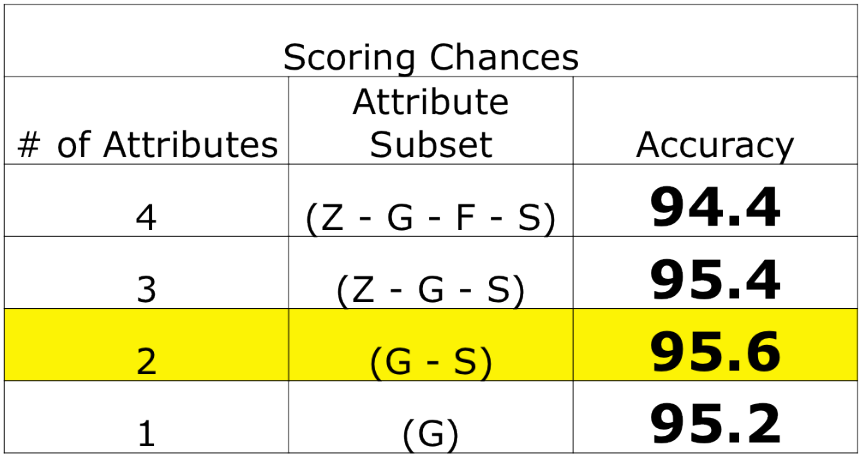
Face-offs		
# of Attributes	Attribute Subset	Accuracy
4	(Z - G - F - S)	**96.9**
3	(G - F - S)	**97.8**
2	(G - F)	**98.3**
1	(F)	**97.1**

Figure 18: We evaluate the best subset of attributes for executing boolean testing for face-offs using k-nearest neighbor classification. The *Direction* attribute is the most important attribute for classification, and the best subset of attributes includes the *Grid-Cell* attribute and the *Direction* attribute.



Breakouts		
# of Attributes	Attribute Subset	Accuracy
4	(Z - G - F - S)	**90.2**
3	(Z - G - F)	**91.7**
2	(G - F)	**92.5**
1	(G)	**92.7**

Figure 19: We evaluate the best subset of attributes for executing boolean testing for breakouts using k-nearest neighbor classification. The *Grid-Cell* attribute is the most important attribute for classification, and the best subset of attributes includes only the the *Grid-Cell* attribute.



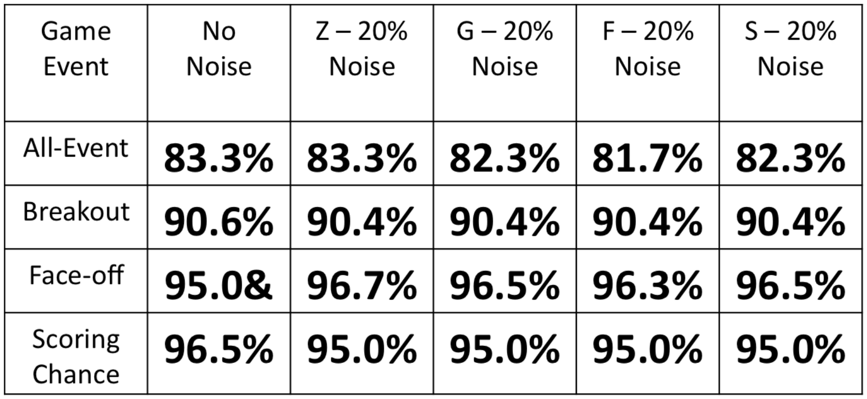
Scoring Chances		
# of Attributes	Attribute Subset	Accuracy
4	(Z - G - F - S)	**94.4**
3	(Z - G - S)	**95.4**
2	(G - S)	**95.6**
1	(G)	**95.2**

Figure 20: We evaluate the best subset of attributes for executing boolean testing for scoring chances using k-nearest neighbor classification. The *Grid-Cell* attribute is the most important attribute for classification, and the best subset of attributes includes the *Grid-Cell* attribute and the *Skating Style* attribute.

collectively. Similar to breakouts, the importance of location information is inherent in the definition of a scoring chance, as a team must be in their offensive zone for the event to occur.

6.5 Introducing Noise

We introduce noise to each of our four player attributes to simulate realistic attribute collection using image processing techniques. Introducing noise randomizes the value a particular attribute for a specified percentage of instances in the dataset. By manually annotating player attribute information from static frames of game play, we have an unrealistically high confidence in the player attribute information that we use for game event classification. We add 20 percent noise to individual attributes and classify game events using decision tree classification. The results for this test can be seen in Figure 21. Adding 20 percent noise to individual attributes does not prove to affect classification accuracy. Classification accuracy for all-event classification and boolean classification of scoring chances falls very slightly when noise is added. Accuracy actually increases when noise is added for boolean classification of face-offs, and the results remain the same for boolean classification of breakouts. It is important to note that adding noise to each attribute for this test is not completely authentic. As described in Section 4.3, we use *NA* to represent the player attributes



Game Event	No Noise	Z – 20% Noise	G – 20% Noise	F – 20% Noise	S – 20% Noise
All-Event	**83.3%**	**83.3%**	**82.3%**	**81.7%**	**82.3%**
Breakout	**90.6%**	**90.4%**	**90.4%**	**90.4%**	**90.4%**
Face-off	**95.0&**	**96.7%**	**96.5%**	**96.3%**	**96.5%**
Scoring Chance	**96.5%**	**95.0%**	**95.0%**	**95.0%**	**95.0%**

Figure 21: We introduce noise to each player attribute individually and use decision tree classification to classify game events. There is not a significant difference between the results found with and without noise added.

that describe players that are not visible in a given frame. For this reason, *NA* is an option during the randomization of attribute values that is done to add noise to a particular attribute. With image processing techniques, we would not have a scenario where visible players have any attribute values that are *NA*, and off-frame players could not have real attribute values.

6.6 Conclusions

We can draw a series of conclusions from the results that we found during the testing process. It is clear that all-event classification is a more difficult task than boolean classification. It would be interesting to evaluate how the addition of new game events of varying classification difficulty could effect this difference. While the contribution of each player attribute to high classification accuracy appears fairly equal in the results shown in Figure 16, it is clear in the subset testing results that the *Grid-Cell* attribute is the most important attribute for high classification results. By increasing the number of grid-cells that cover the playing surface, we can achieve a better understanding of the best location granularity for game event classification. In addition, it is important to note that our dataset consists of an imbalanced collection of frame instances selected arbitrarily from an archive of game film. It would be interesting to explore game event classification in a dataset balanced with a series of game events.

7 Future Work

We hope to expand our system by investigating classification results with more realistic attribute accuracy. By manually annotating player attribute data for each frame instance we have an unrealistically high confidence in the accuracy of each attribute value. It is unlikely that image processing techniques can achieve such a level of confidence in extracting player attribute data from sequences of game film. We can investigate this assumption by running more simulations with varying additions of noise, and comparing the results with classification tests using data gathered by existing image processing techniques. In addition we will investigate how other game events fit into our classification task. By adding more game events we can start to understand what aspects of the game are most fit for classification, and we can use this information to improve our set of attributes used for classification. Further we can investigate how applicable our classification system will be for other sports. We predict that it could prove to be a useful tool for other free-flowing sports like soccer or lacrosse, however static sports like baseball and tennis would not benefit from using this tool. We could also investigate if our classification system is applicable for purposes outside of the field of video analysis in sports.

References

[1] Wongun Choi and Silvio Savarese. Multiple target tracking in world coordinate with single, minimally calibrated camera. In *Proceedings of the 11th European conference on Computer vision: Part IV*, ECCV'10, pages 553–567, Berlin, Heidelberg, 2010. Springer-Verlag.

[2] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21 –27, january 1967.

[3] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 14th international joint conference on Artificial intelligence - Volume 2*, IJCAI'95, pages 1137–1143, San Francisco, CA, USA, 1995. Morgan Kaufmann Publishers Inc.

[4] Wei-Lwun Lu, Kenji Okuma, and James J. Little. Tracking and recognizing actions of multiple hockey players using the boosted particle filter. *Image Vision Comput.*, 27(1-2):189–205, January 2009.

[5] Wei lwun Lu and James J. Little. Tracking and recognizing actions at a distance. In *in: ECCV Workshop on Computer Vision Based Analysis in Sport Environments*, pages 49–60, 2006.

[6] Thair Nu Phyu. Survey of classification techniques in data mining. In *Proceedings of the International MultiConference of Engineers and Computer Scientists 2009 Vol I*, 2009.

[7] Christof Ridder, Olaf Munkelt, and Harald Kirchner. Adaptive background estimation and foreground detection using kalman-filtering. pages 193–199, 1995.

[8] I. Rish. An empirical study of the naive bayes classifier. In *IJCAI 2001 workshop on empirical methods in artificial intelligence*, volume 3, pages 41–46, 2001.

[9] S.R. Safavian and D. Landgrebe. A survey of decision tree classifier methodology. *Systems, Man and Cybernetics, IEEE Transactions on*, 21(3):660 –674, may/jun 1991.

[10] J. R. Wang and N. Parameswaran. Survey of sports video analysis: research issues and applications. In *Proceedings of the Pan-Sydney area workshop on Visual information processing*, VIP '05, pages 87–90, Darlinghurst, Australia, Australia, 2004. Australian Computer Society, Inc.

[11] Mike Willard. Sportscode gamebreaker. In *Sportscode Gamebreaker*, Camarillo, CA, 2013.

[12] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), December 2006.

[13] Sunghoon Choi Yongduek, Sunghoon Choi, Yongduek Seo, Hyunwoo Kim, and Ki sang Hong. Where are the ball and players? : Soccer game analysis with color-based tracking and image mosaick. In *Proc. ICIAP*, pages 196–203, 1997.