Problem statement

Good job explaining the motivation behind your project! Regarding your comment about accuracy vs. recall, you might want to compare your results using ROC curves.

http://scikit-learn.org/stable/modules/model_evaluation.html#roc-metrics
https://www.youtube.com/watch?v=OAl6eAyP-yo

Breast cancer data set from UCI repository

It's a good idea to start your analysis with a complementary data set to learn more about breast cancers and gain some "expert knowledge" that will help you later in your analysis of the MIAS data set. For instance, you might want to use the "Breast Cancer Wisconsin (Diagnostic) Data Set" from the UCI ML repository which contains data about the radius, texture, perimeter, area, smoothness of breast cancer tumors and their type: malignant or benign. https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+(Diagnostic)

For instance, you could use this data set to better understand the relationship between those features and define some of the characteristics of malignant tumors. Here are some ideas.

- Plot the data points using PCA. Analyze and plot the principal components using a Biplot. Mark malignant and benign points with different colors.
- Build a decision tree to learn the important features for detecting malignant tumors. Plot it using Graphviz.
- Fit SVM classifiers to detect malignant tumors based on this data. Try with a linear and an RBF kernel. Analyze the results using ROC curves.

Exploratory data analysis

Before applying the machine learning algorithms it could be interesting to include some exploratory and qualitative analysis of the UCI and MIAS data sets. For instance, you might want to

- Create visualizations that give some preliminary insights into the data (i.e. histograms of tumor radius)
- Investigate and plot relationships between various tumour characteristics (from the UCI dataset) such as radius, perimeter, texture, etc.
- Create a gallery of tumors for the MIAS data set, organized by types of abnormality.

Create an alternative data set

As you explained in your proposal, it might be difficult to detect the abnormalities on the 64x64 low-resolution images, and working on the high-res ones is computationally expensive. So, how about creating an alternative, simpler data set with 64x64 patches from the high-res images? If the image contains an abnormality, then extract the 64x64 patch centered at the abnormality, otherwise, pick a random 64x64 patch. That way, you can start your analysis by building a binary normal-abnormal classifier and a benign-malignant one for this set of small images.

If you think that it makes sense to create this alternative data set, please update your proposal.

Backup plan

The MIAS data set is challenging for several reasons.

- The size of the data set is relatively small: you only have 300 scans.
- The type of images: you might use pre-trained networks to extract features (transfer learning), but these are usually trained on the ImageNet data set which is relatively different from this one, i.e., the features used to detect cat and dog breeds might be irrelevant to detect breast cancer tumors.

Hence, it's important that you list in your proposal the existing results that you can find online. Also, can you include more details on how you plan to use pre-trained networks like OverFeat, AlexNet or VGG? For instance, search for code examples on GitHub or tutorials online. You might want to take a look at http://cs231n.github.io/transfer-learning/

Also, it would be good if you have a backup plan in case you don't get good results with transfer learning on the MIAS data set. For instance, did you look at the DDSM data set mentioned in the paper from your proposal? It might have enough images to train a simple ConvNet.

Paper https://arxiv.org/pdf/1612.00542.pdf
DDSM website http://marathon.csee.usf.edu/Mammography/Database.html

Additional resources

You might want to check the following links.

"Density-Wise Two Stage Mammogram Classification using Texture Exploiting Descriptors"
https://arxiv.org/pdf/1701.04010.pdf
"Transfer learning in tensorflow using a pre-trained inception-resnet-v2 model"
https://kwotsin.github.io/tech/2017/02/11/transfer-learning.html